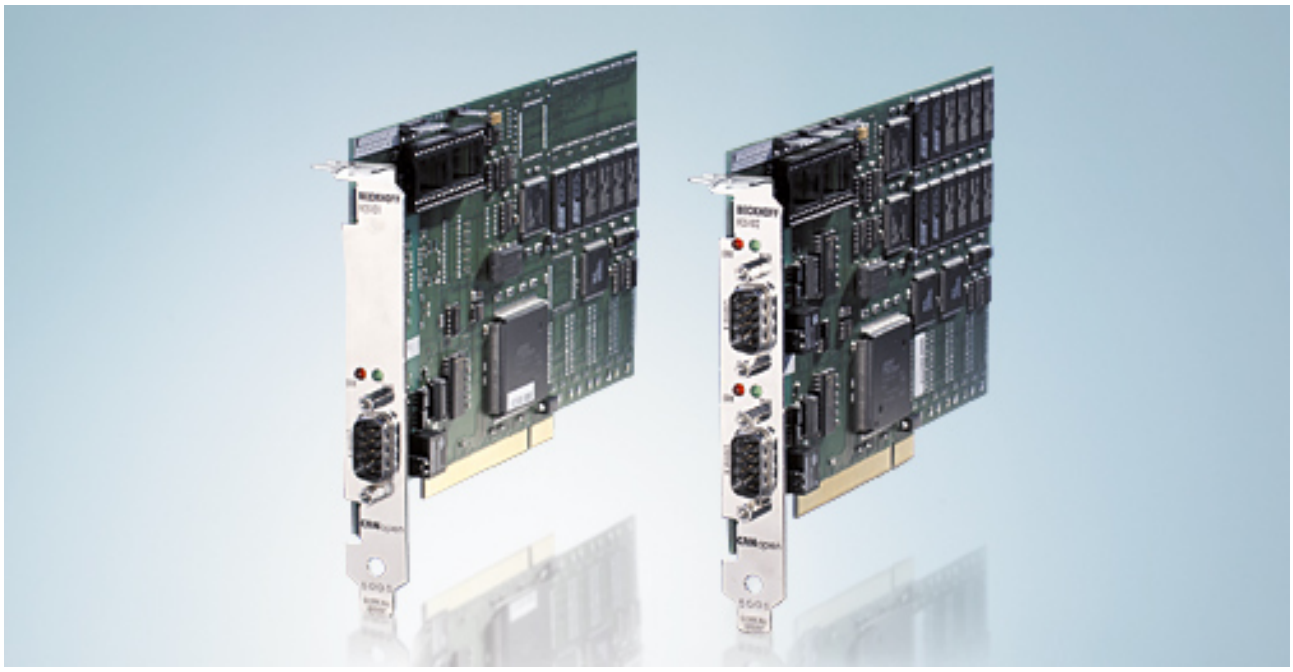


Dokumentation

FC5101 und FC5102

PCI-Karten für CANopen

Version: 2.0
Datum: 17.11.2017

BECKHOFF

Inhaltsverzeichnis

1 Vorwort

1.1 Hinweise zur Dokumentation

Zielgruppe

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.
Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.
Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentliche Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.
Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.
Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC® und XTS® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.
Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente: EP1590927, EP1789857, DE102004044764, DE102007017835 mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

Die TwinCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente: EP0851348, US6167425 mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizensiert durch die Beckhoff Automation GmbH, Deutschland.

Copyright

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Erklärung der Hinweise

In der vorliegenden Dokumentation werden die folgenden Hinweise verwendet.
Diese Hinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

⚠ GEFAHR

Akute Verletzungsgefahr!

Wenn dieser Sicherheitshinweis nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

⚠ WARNUNG

Verletzungsgefahr!

Wenn dieser Sicherheitshinweis nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

⚠ VORSICHT

Schädigung von Personen!

Wenn dieser Sicherheitshinweis nicht beachtet wird, können Personen geschädigt werden!

HINWEIS

Schädigung von Umwelt/Geräten oder Datenverlust

Wenn dieser Hinweis nicht beachtet wird, können Umweltschäden, Gerätebeschädigungen oder Datenverlust entstehen.

Tipp oder Fingerzeig

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

1.3 Ausgabestände der Dokumentation

Version	Kommentar
2.0	• Migration
1.0	• Vollständig überarbeitet ◦ FC510x Monitor Software dokumentiert ◦ CANopen Protokollbeschreibung überarbeitet
0.9 (Pre-Release)	• Vorab-Version, Stand:11.03.2002

Der Einsatz der FC5101 im Slave Modus ist in einer separaten Dokumentation (FC510x Slave.chm bzw. -.pdf) beschrieben.

2 Produktübersicht

2.1 Hardware Übersicht

CAN-Abschlusswiderstand

Auf der Karte sind CAN Abschlusswiderstände (120 Ohm) vorhanden. Diese können über einen Jumper (bis Hardwarestand 3) bzw. Schiebeschalter (ab Hardwarestand 4) nahe der CAN Stecker aktiviert werden.

Der Flash Disk Sockel ist unbestückt.

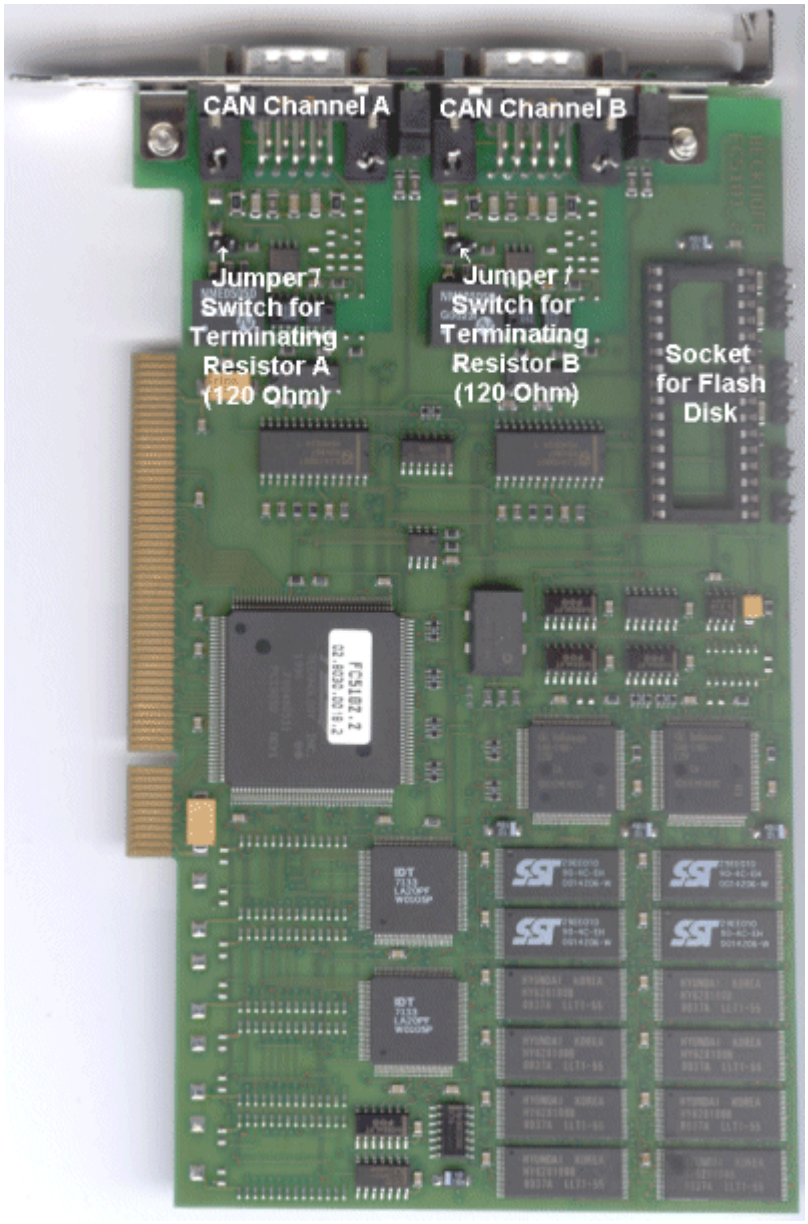


Abb. 1: FC5102

Steckerbelegung

Die CAN Busleitung wird über 9polige Sub-D-Buchsen mit folgender Steckerbelegung angeschlossen.

Pin	Belegung
2	CAN low (CAN-)
3	CAN Ground (intern verbunden mit Pin 6)
5	Schirm
6	CAN Ground (intern verbunden mit Pin 3)
7	CAN high (CAN+)

Die nicht aufgeführten Pins sind nicht verbunden.

Hinweis: an Pin 9 darf eine Hilfsspannung bis 30 V_{DC} angeschlossen sein (wird von manchen CAN Geräten zur Versorgung der Transceiver genutzt).

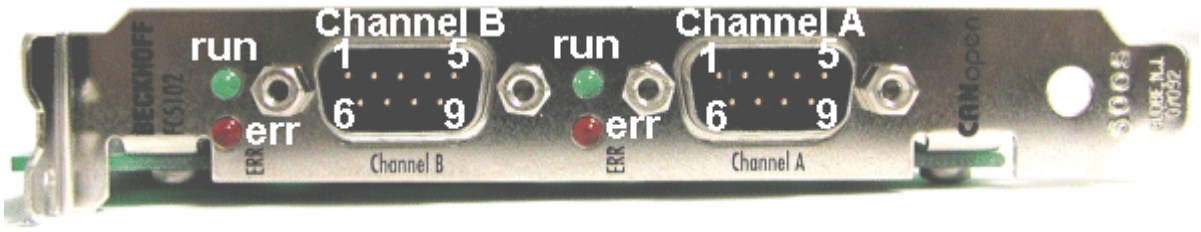


Abb. 2: FC5102Blende

2.2 Technische Daten

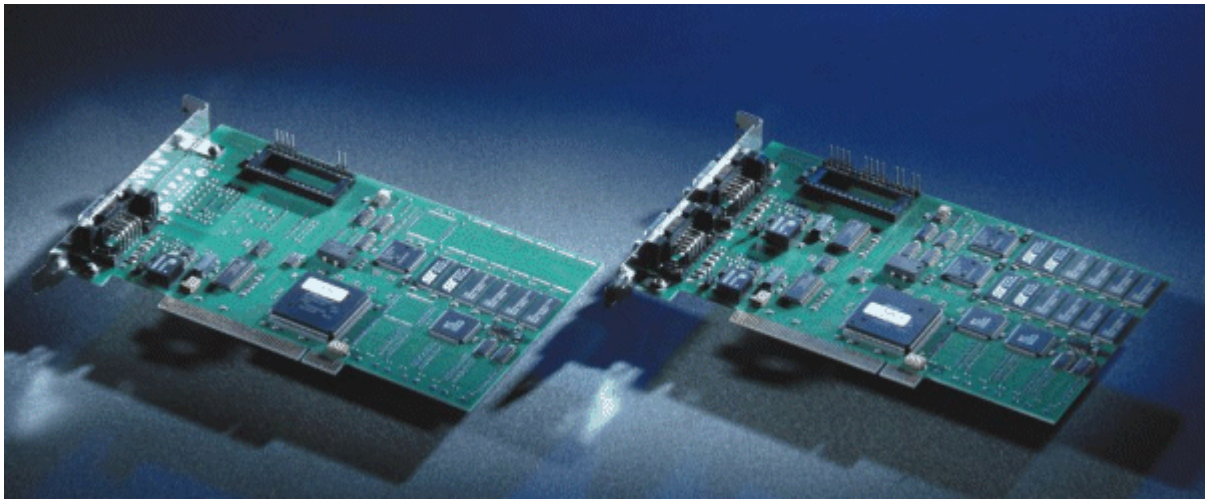


Abb. 3: FC510x

Die FC510x ist eine CANopen [▶ 11] Masterkarte mit folgenden Eigenschaften:

- ein (FC5101) oder zwei CAN-Kanäle (FC5102), jeweils mit eigenem Prozessor, Speicher etc.
- Wahlweise CANopen Master oder Slave.
- Alle PDO Kommunikationsarten werden unterstützt.
- Prozessabbild: Summe max. 3 kBytes Input- und Outputdaten.
- Jedes PDO kann individuell überwacht werden.
- Host-Kommunikation kann freilaufend, synchronisiert oder äquidistant sein.
- Äquidistant-Modus für Antriebsregelung über den Bus: SYNC Objekte werden im Mittel quarzgenau übertragen, Prozessdatenaustausch zur Applikation ist durchsynchronisiert (nur mit TwinCAT).
- Emergency Nachrichten werden von der Karte gespeichert.
- Error Handling kann für jeden Busknoten individuell eingestellt werden.
- allgemeine CAN Nachrichten (CAN Schicht 2) können gesendet und empfangen werden.
- Leistungsfähiges Parameter- und Diagnoseinterface.
- Integrierte Buslast-Anzeige.
- CAN Schnittstellen sind galvanisch getrennt.
- Entspricht CANopen Spezifikation DS301 V4.01.
- Boot-Up nach DS302.
- Treiber: TwinCAT I/O für WinNT, Win2k, WinXP;
- Driver Construktion Kit für andere Betriebssysteme auf Anfrage.

Details zur Slave Funktionalität siehe entsprechende separate Dokumentation.

2.3 CANopen Einführung

Abb. 4: CANopenLogo

CANopen ist eine weit verbreitete CAN-Anwendungsschicht, die im Verband CAN-in-Automation (CiA, http://www.can-cia.org) entwickelt und inzwischen zur internationalen Normung angenommen wurde.

Gerätemodell

CANopen besteht aus der Protokolldefinition (Kommunikationsprofil) sowie den Geräteprofilen, die den Dateninhalt für die jeweilige Geräteklasse normieren. Zur schnellen Kommunikation der Ein- und Ausgangsdaten dienen die Prozessdatenobjekte (PDO) [▶ 45]. Die CANopen-Geräteparameter und Prozessdaten sind in einem Objektverzeichnis strukturiert. Der Zugriff auf beliebige Daten dieses Objektverzeichnisses erfolgt über die Servicedatenobjekte (SDO). Weiter gibt es einige Spezialobjekte (bzw. Telegrammarten) für Netzwerkmanagement (NMT), Synchronisation, Fehlermeldungen etc.

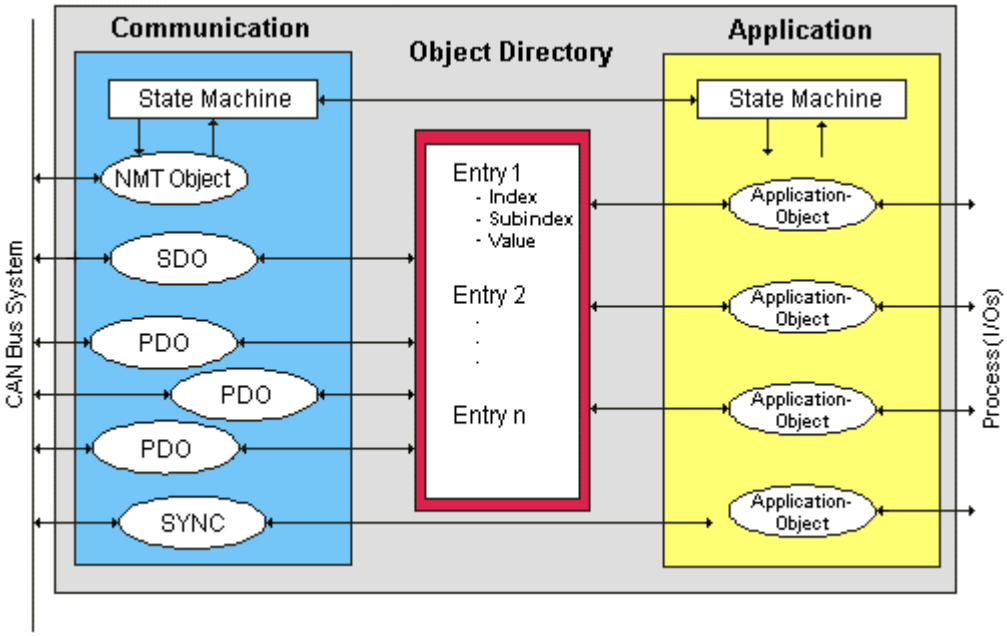


Abb. 5: CANopen Gerätemodell

Kommunikationsarten

CANopen definiert mehrere Kommunikationsarten für die Ein- und Ausgangsdaten (Prozessdatenobjekte):

- Ereignisgesteuert [▶ 47]: Telegramme werden versendet, sobald sich der Inhalt geändert hat. Hier wird nicht ständig das Prozessabbild, sondern nur die Änderung desselben übertragen.
- Zyklisch synchron [▶ 47]: Über ein SYNC Telegramm werden die Baugruppen veranlasst, die vorher empfangenen Ausgangsdaten zu übernehmen und neue Eingangsdaten zu senden.
- Angefordert (gepollt) [▶ 45]: Über ein CAN Datenanforderungstelegramm werden die Baugruppen veranlasst ihre Eingangsdaten zu senden.

Die gewünschte Kommunikationsart wird über den Parameter Transmission Type [▶ 45] eingestellt.

Geräteprofil

Die BECKHOFF CANopen-Geräte unterstützen alle E/A- Kommunikationsarten und entsprechen dem Geräteprofil für digitale und analoge Ein-/Ausgabebaugruppen (DS401 Version 1). Aus Gründen der Abwärtskompatibilität wurde das Default Mapping nicht der Profilversion DS401 V2 angepasst.

Übertragungsraten

Neun Übertragungsraten von 10 kBaud bis 1 MBaud stehen für unterschiedliche Buslängen zur Verfügung. Durch die effektive Nutzung der Busbandbreite erreicht CANopen kurze Systemreaktionszeiten bei vergleichsweise niedrigen Datenraten.

Topologie

CAN basiert auf einer linienförmigen Topologie [▶ 14]. Die Anzahl der Teilnehmer pro Netz ist dabei von CANopen logisch auf 128 begrenzt, physikalisch erlaubt die aktuelle Treiber-Generation bis zu 64 Knoten in einem Netzsegment. Die bei einer bestimmten Datenrate maximal mögliche Netzausdehnung ist durch die auf dem Busmedium erforderliche Signallaufzeit begrenzt. Bei 1 MBaud ist z. B. eine Netzausdehnung von 25 m, bei 50 kBaud eine Netzausdehnung von 1000 m möglich. Bei niedrigen Datenraten kann die Netzausdehnung durch den Einsatz von Repeatern erhöht werden, diese ermöglichen auch den Aufbau von Baumstrukturen.

Buszugriffsverfahren

CAN arbeitet nach dem Verfahren Carrier Sense Multiple Access (CSMA), d.h. jeder Teilnehmer ist bezüglich des Buszugriffs gleichberechtigt und kann auf den Bus zugreifen, sobald dieser frei ist (Multi-Master-Buszugriff). Der Nachrichtenaustausch ist dabei nicht Teilnehmerbezogen sondern Nachrichtenbezogen. Das bedeutet, dass jede Nachricht mit einem priorisierten Identifier eindeutig gekennzeichnet ist. Damit beim Verschicken der Nachrichten verschiedener Teilnehmer keine Kollisionen auf dem Bus entstehen, wird beim Start der Datenübertragung eine bitweise Busarbitrierung durchgeführt. Die Busarbitrierung vergibt die Busbandbreite an die Nachrichten in der Reihenfolge ihrer Priorität, am Ende der Arbitrierungsphase belegt jeweils nur ein Busteilnehmer den Bus, Kollisionen werden vermieden und die Bandbreite wird optimal genutzt.

Konfiguration und Parametrierung

Mit dem TwinCAT System Manager können alle CANopen Parameter komfortabel eingestellt werden. Für die Parametrierung der Beckhoff CANopen-Geräte mit Konfigurationstools dritter Hersteller steht Ihnen auf der Beckhoff Website (http://www.beckhoff.de) ein eds-File (electronic data sheet) zur Verfügung.

Zertifizierung

Die Beckhoff CANopen-Geräte verfügen über eine leistungsfähige Protokollimplementierung und sind vom Verband CAN-in-Automation (http://www.can-cia.org) zertifiziert.

3 Einbau und Verdrahtung

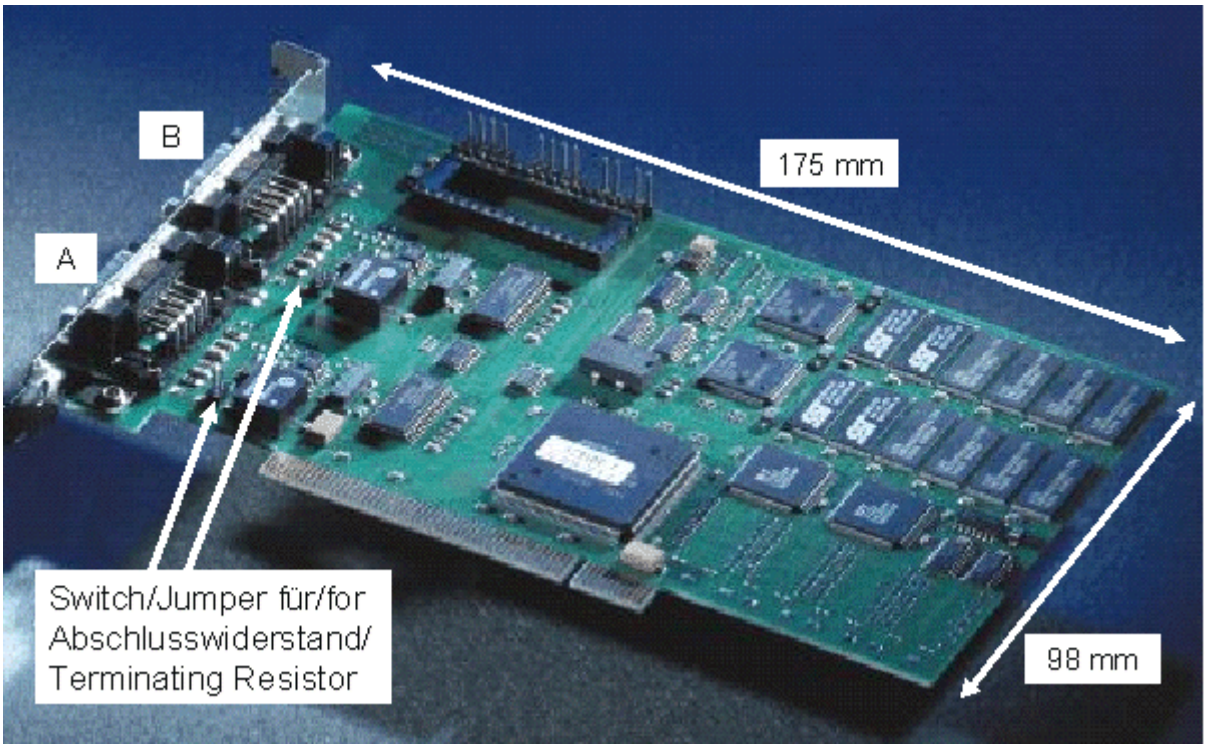
3.1 Einbau

HINWEIS

Qualifizierter Einbau

Der Einbau der Feldbus-PCI-Karten darf nur von qualifiziertem Personal unter Beachtung der folgenden Hinweise vorgenommen werden.

- Zum Schutz der Karte vor Entladung mit statischer Elektrizität muss sich das Bedienpersonal vor dem Berühren der Karte oder des PCs elektrostatisch entladen.
- Vor dem Öffnen des PC-Gehäuses muss das Gerät ausgeschaltet und der Netzstecker abgezogen werden.



Vor dem Einbau sind ggf. die Jumper zur Aktivierung der internen CAN-Busabschluss-Widerstände zu setzen bzw. die Schalter einzuschalten (ab HW-Stand 3). Jumper gesetzt bzw. Schalter ein [ON] bedeutet hierbei: Abschlusswiderstand zugeschaltet.

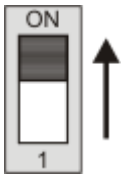


Abb. 6: DiP-Schalter ON = Abschlusswiderstand zugeschaltet

Die Karte kann in einen beliebigen freien PCI Slot eingebaut werden. Achten Sie beim Einbau auf eine gute Kontaktierung des PCI Bussteckers und festen Sitz der Baugruppe. Fixieren Sie die Baugruppe mit der Befestigungsschraube am PC Slot Gehäuse.

3.2 CANopen Verkabelung

Hinweise für die Überprüfung der CAN-Verdrahtung finden sich im Kapitel Fehlersuche / Trouble Shooting [▶ 72].

3.2.1 CAN-Topologie

CAN ist ein 2-Draht-Bussystem, an dem alle Teilnehmer parallel (d.h. mit kurzen Stichleitungen) angeschlossen werden. Der Bus muss an jedem Ende mit einem Abschlusswiderstand von 120 (bzw. 121) Ohm abgeschlossen werden, um Reflexionen zu vermeiden. Dies ist auch bei sehr kurzen Leitungslängen erforderlich!

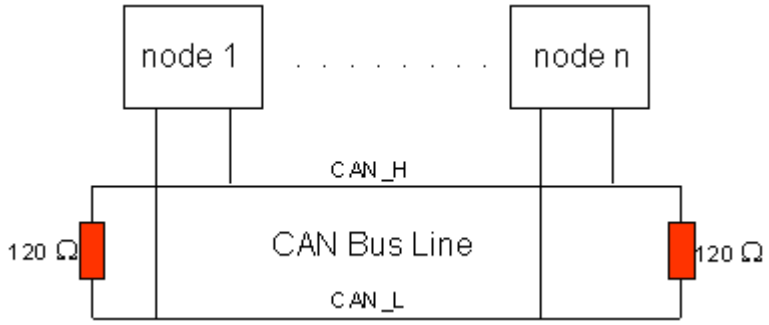


Abb. 7: Abschluss des Busses mit Abschlusswiderstand 120 Ohm

Da die CAN-Signale als Differenzpegel auf dem Bus dargestellt werden, ist die CAN-Leitung vergleichsweise unempfindlich gegen eingeprägte Störungen (EMI). Es sind jeweils beide Leitungen betroffen, somit verändert die Störung den Differenzpegel kaum.

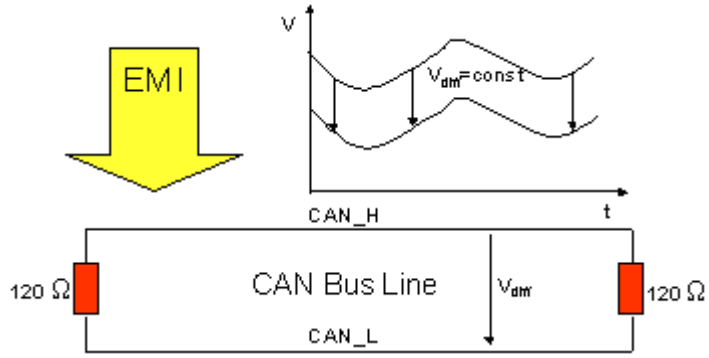


Abb. 8: Unempfindlichkeit gegen eingeprägte Störungen

3.2.2 Buslänge

Die maximale Buslänge wird bei CAN vorwiegend durch die Signallaufzeit beschränkt. Das Multi-Master-Buszugriffsverfahren (Arbitrierung) erfordert, dass die Signale quasi gleichzeitig (vor der Abtastung innerhalb einer Bitzeit) an allen Knoten anliegen. Da die Signallaufzeit in den CAN-Anschaltungen (Transceiver, Optokoppler, CAN-Controller) nahezu konstant sind, muss die Leitungslänge an die Baud-Rate angepasst werden.

Baud-Rate	Buslänge
1 MBit/s	< 20 m*
500 kBit/s	< 100 m
250 kBit/s	< 250 m
125 kBit/s	< 500 m
50 kBit/s	< 1000 m
20 kBit/s	< 2500 m
10 kBit/s	< 5000 m

*) Häufig findet man in der Literatur für CAN die Angabe 40 m bei 1 MBit/s. Dies gilt jedoch nicht für Netze mit optoentkoppelten CAN-Controllern. Die worst case Berechnung mit Optokopplern ergibt bei 1 MBit/s eine maximale Buslänge von 5 m - erfahrungsgemäß sind jedoch 20 m problemlos erreichbar.

Bei Buslängen über 1000 m kann der Einsatz von Repeatern notwendig werden.

3.2.3 Stichleitungen

Stichleitungen ("drop lines") sind nach Möglichkeit zu vermeiden, da sie grundsätzlich zu Signalreflexionen führen. Die durch Stichleitungen hervorgerufenen Reflexionen sind jedoch in der Regel unkritisch, wenn sie vor dem Abtastzeitpunkt vollständig abgeklungen sind. Bei den in den Buskopplern gewählten Bit-Timing-Einstellungen kann dies angenommen werden, wenn folgende Stichleitungslängen nicht überschritten werden:

Baud-Rate	Länge Stichleitung	gesamte Länge aller Stichleitungen
1 MBit/s	< 1 m	< 5 m
500 kBit/s	< 5 m	< 25 m
250 kBit/s	< 10 m	< 50 m
125 kBit/s	< 20 m	< 100 m
50 kBit/s	< 50 m	< 250 m

Stichleitungen dürfen nicht mit Abschlusswiderständen versehen werden.

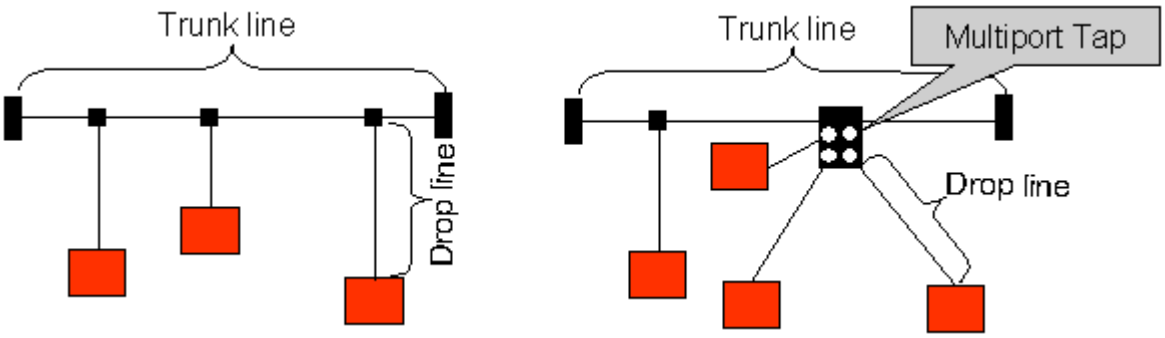


Abb. 9: Beispieltopologie Stichleitungen

3.2.4 Sternverteiler (Multiport Tap)

Beim Einsatz von passiven Verteilern ("Multiport Taps"), z. B. der Beckhoff Verteilerbox ZS5052-4500 sind kürzere Stichleitungslängen einzuhalten. Die folgende Tabelle gibt die maximalen Stichleitungslängen und die maximale Länge der Trunk Line (ohne Stichleitungen) an:

Baud-Rate	Länge Stichleitung bei Multi-port Topologie	Länge Trunk Line (ohne Stichleitungen)
1 MBit/s	< 0,3 m	< 25 m
500 kBit/s	< 1,2 m	< 66 m
250 kBit/s	< 2,4 m	< 120 m
125 kBit/s	< 4,8 m	< 310 m

3.2.5 CAN-Kabel

Für die CAN-Verdrahtung wird die Verwendung von paarig verdrillten, geschirmten Kabeln (2x2) mit einem Wellenwiderstand von 108...132 Ohm empfohlen. Wenn das Bezugspotential der CAN-Transceiver (CAN-Ground) nicht verbunden werden soll, so kann auf das zweite Adernpaar verzichtet werden (nur bei kleinen Netzausdehnungen mit gemeinsamer Speisung aller Teilnehmer empfehlenswert).

ZB5100 CAN-Kabel

Beckhoff hat ein hochwertiges CAN-Kabel mit folgenden Eigenschaften im Programm:

- 2 x 2 x 0,25 mm² (AWG 24) paarig verseilt, Kabelfarben: rot/schwarz + weiß/schwarz
- doppelt geschirmt
- Schirmgeflecht mit Beilauflitze (kann direkt auf Pin3 der 5-pol Anschlussklemme aufgelegt werden)
- flexibel (Mindestbiegeradius 35 mm bei einmaligem Biegen, 70 mm bei mehrmaligem Biegen)
- Wellenwiderstand (60 kHz): 120 Ohm
- Leiterwiderstand < 80 Ohm/km
- Mantel: PVC grau, Außendurchmesser 7,3 +/- 0,4 mm
- Gewicht: 64 kg/km.
- Bedruckt mit "Beckhoff ZB5100 CAN-BUS 2x2x0.25" und Metermarkierung (Längenangaben, alle 20cm)

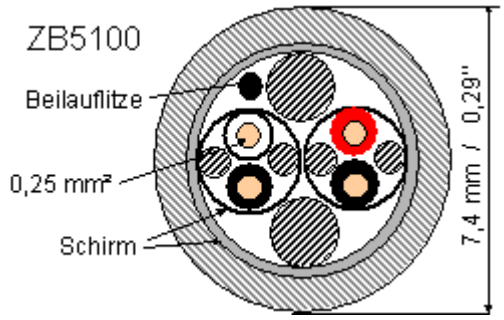


Abb. 10: Aufbau CAN-Kabel ZB5100

ZB5200 CAN/DeviceNet-Kabel

Das Kabelmaterial ZB5200 entspricht der DeviceNet Spezifikation und eignet sich ebenso für CANopen Systeme. Aus diesem Kabelmaterial sind auch die vorkonfektionierten Busleitungen ZK1052-xxxx-xxxx für die Feldbus Box Baugruppen gefertigt. Es hat folgende Spezifikation:

- 2 x 2 x 0,34 mm² (AWG 22) paarig verseilt
- doppelt geschirmt, Schirmgeflecht mit Beilauflitze
- Wellenwiderstand (1 MHz): 126 Ohm
- Leiterwiderstand 54 Ohm/km
- Mantel: PVC grau, Außendurchmesser 7,3 mm
- Bedruckt mit "InterlinkBT DeviceNet Type 572" sowie UL und CSA Ratings
- Litzenfarben entsprechen DeviceNet Spezifikation
- UL anerkanntes AWM Type 2476 Rating

- CSA AWM I/II A/B 80°C 300V FT1
- Entspricht DeviceNet "Thin Cable" Spezifikation

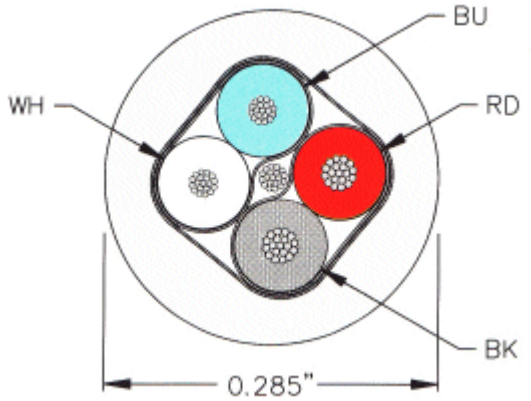


Abb. 11: Aufbau CAN-/DeviceNet-Kabel ZB5200

3.2.6 Schirmung

Der Schirm ist über das gesamte Buskabel zu verbinden und nur an einer Stelle galvanisch zu erden um Masseschleifen zu vermeiden.
Das Schirmungskonzept mit HF-Ableitung von Störungen über R/C-Glieder auf die Tragschiene geht davon aus, dass die Tragschiene entsprechend geerdet und störungsfrei ist. Sollte dies nicht der Fall sein, so kann es vorkommen, dass HF-Störpegel über die Tragschiene auf den Schirm des Buskabels übertragen werden. In diesem Fall sollte der Schirm an den Kopplern nicht aufgelegt werden - aber dennoch komplett durchverbunden sein.

Hinweise für die Überprüfung der CAN-Verdrahtung finden sich im Kapitel Fehlersuche / Trouble Shooting [▶ 72].

3.2.7 Kabelfarben

Vorschlag für die Verwendung der Beckhoff CAN-Kabel an Busklemme und Feldbus Box:

Pin BK51x0 PIN BX5100 (X510)	Pin BK5151 CX8050, CX8051, CXxxxx-B510/M510	Pin Feld-bus Box	Pin FC51xx	Funktion	Kabelfarbe ZB5100	Kabelfarbe ZB5200
1	3	3	3	CAN Ground	**schwarz** /(rot)	**schwarz**
2	2	5	2	CAN Low	**schwarz**	**blau**
3	5	1	5	Schirm	Beilauflitze	Beilauflitze
4	7	4	7	CAN high	**weiß**	**weiß**
5	9	2	9	nicht benutzt	**(rot)**	**(rot)**

3.2.8 BK5151, FC51xx, CX mit CAN Interface und EL6751: D-Sub 9polig

Die CAN Busleitung wird an die FC51x1, FC51x2 CANopen Karten und bei der EL6751 CANopen Master-/Slaveklemme über 9polige Sub-D-Buchsen mit folgender Steckerbelegung angeschlossen.

Pin	Belegung
2	CAN low (CAN-)
3	CAN Ground (intern verbunden mit Pin 6)
6	CAN Ground (intern verbunden mit Pin 3)
7	CAN high (CAN+)

Die nicht aufgeführten Pins sind nicht verbunden.
Die Tragschienenkontaktfeder und der Steckerschirm sind durchverbunden.

Hinweis: an Pin 9 darf eine Hilfsspannung bis 30 V_{DC} angeschlossen sein (wird von manchen CAN Geräten zur Versorgung der Transceiver genutzt).

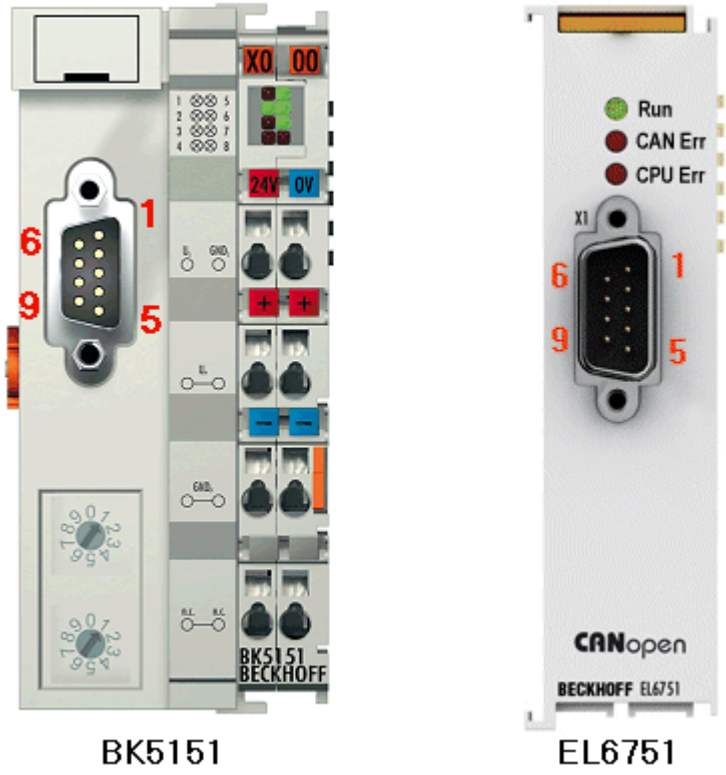


Abb. 12: Pinbelegung BK5151, EL6751

FC51x2:

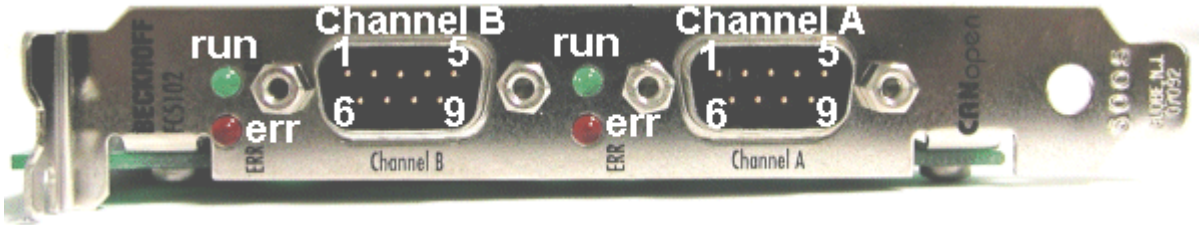


Abb. 13: FC51x2

3.2.9 BK51x0/BX5100: 5poliger Open Style Connector

Bei den BK51x0/BX5100 (X510) Buskopplern befindet sich auf der linken Seite eine abgesenkte Frontfläche mit einem 5poligen Stecker.
Hier kann die mitgelieferte CANopen- Verbindungsbuchse eingesteckt werden.

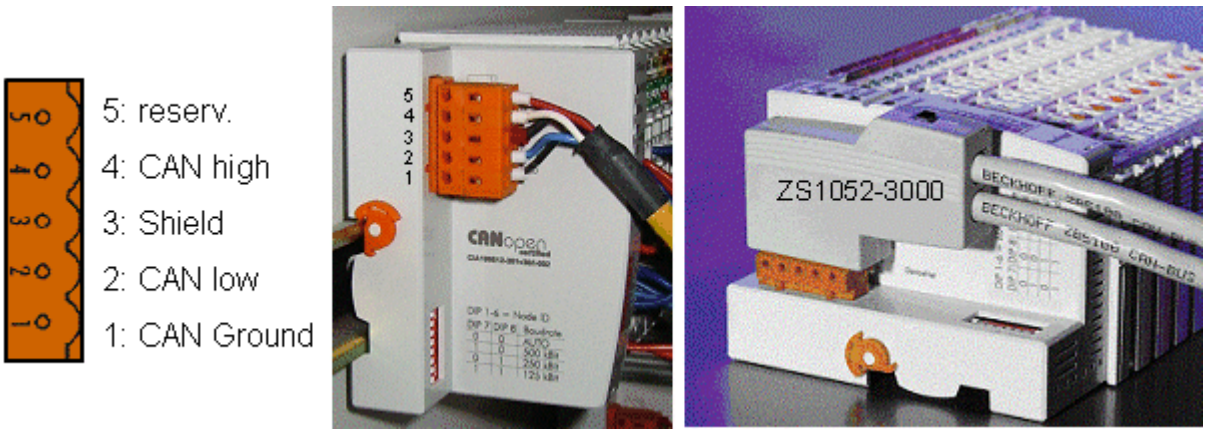


Abb. 14: Belegung Verbindungsbuchse BK51x0/BX5100

Das linke Bild zeigt die Buchse im Buskoppler BK51x0/BX5100. Pin 5 ist dabei der oberste Pin auf der Steckerleiste. Pin 5 ist nicht benutzt. Pin 4 ist die CAN-High-Leitung, Pin 2 die CAN-Low-Leitung und an Pin 3 wird der Schirm aufgelegt (ist über eine R/C-Schaltung mit der Tragschiene verbunden). Optional kann am Pin 1 CAN-GND angeschlossen werden. Wenn alle CAN-Ground Pins verbunden sind ergibt dies ein gemeinsames Bezugspotential für die CAN Transceiver im Netz. Es empfiehlt sich, CAN-GND an einer Stelle zu erden, damit das gemeinsame CAN Bezugspotential nahe beim Versorgungs-Potential liegt. Da die CANopen Buskoppler BK51X0/BX5100 über eine vollständige galvanische Trennung des Busanschlusses verfügen, kann u.U. auf die Verdrahtung von CAN-Ground verzichtet werden.

Businterface Connector ZS1052-3000

Alternativ zum mitgelieferten Stecker kann der CAN Interface Connector ZS1052-3000 eingesetzt werden. Dieser vereinfacht die Verdrahtung erheblich. Für die ankommenden und abgehenden Leitungen stehen separate Klemmen zur Verfügung, der Schirm wird durch die Zugentlastung großflächig angeschlossen. Der integrierte Abschlusswiderstand kann von außen zugeschaltet werden. Ist er eingeschaltet, so wird die abgehende Busleitung elektrisch abgetrennt - damit lassen sich Verdrahtungsfehler schnell lokalisieren, und es ist gewährleistet, dass nicht mehr als zwei Widerstände im Netz aktiv sind.

3.2.10 LC5100: Busanschluss über Federkraftklemmen

Beim Low-Cost-Koppler LC5100 wird die CAN-Leitung direkt auf die Klemmstellen 1 (CAN-H, gekennzeichnet mit C+) bzw. 5 (CAN-L, gekennzeichnet mit C-) aufgelegt. Der Schirm kann optional auf die Klemmstellen 4 bzw. 8 aufgelegt werden, diese sind über eine R/C-Schaltung mit der Tragschiene verbunden.

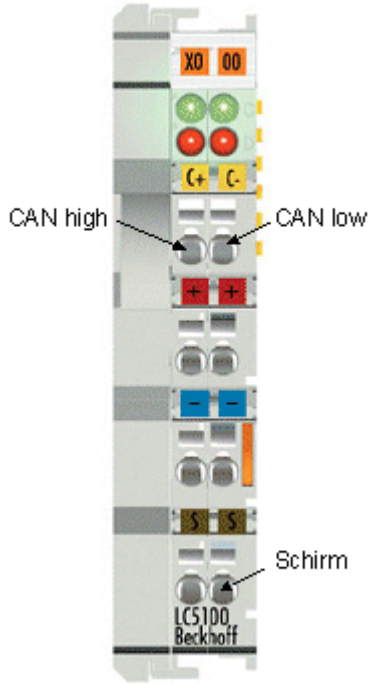


Abb. 15: LC5100

HINWEIS

Beschädigung der Geräte möglich!

Durch die nicht vorhandene galvanische Trennung kann durch unsachgemäße Verkabelung der CAN Treiber zerstört oder geschädigt werden. Verkabeln sie immer im ausgeschalteten Zustand.
Verkabeln Sie erst die Spannungsversorgung und legen sie erst dann den CAN auf. Überprüfen Sie die Verkabelung und schalten dann erst die Spannung ein.

3.2.11 Feldbus Box: M12 CAN Buchse

Bei der Feldbus Box IPxxxx-B510, IL230x-B510 und IL230x-C510 wird der Busanschluss mit 5poligen M12 Steckverbindern ausgeführt.

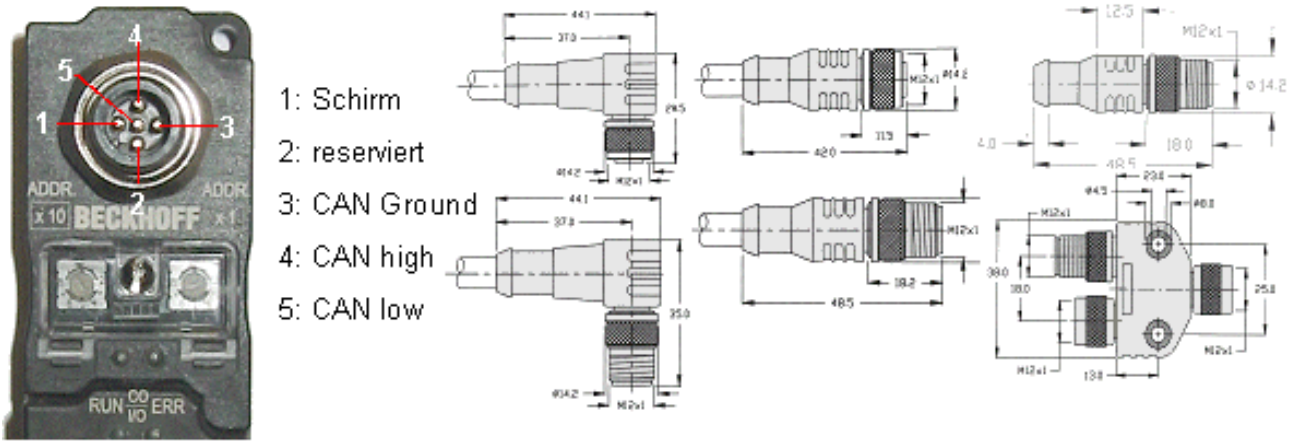


Abb. 16: Pinbelegung M12 Stecker Feldbus Box

Für das Feldbus Box System bietet Beckhoff feldkonfektionierbare Stecker, Passivverteiler, Abschlusswiderstände sowie eine große Auswahl an vorkonfektionierten Kabeln an. Details finden sich im Katalog oder unter www.beckhoff.de.

4 Parametrierung und Inbetriebnahme

4.1 Konfiguration: TwinCAT System Manager

Zur Konfiguration der FC510x CANopen PCI Karte dient das TwinCAT System Manager Tool. Der System Manager stellt die Anzahl und Programme der TwinCAT SPS-Systeme, die Konfiguration der Achsregelung und die angeschlossenen E/A-Kanäle als Struktur dar und organisiert die Mappings des Datenverkehrs.

Abb. 17: TwinCAT System Manager

Für Applikationen ohne TwinCAT SPS oder NC konfiguriert das TwinCAT System Manager Tool die Programmierschnittstellen für vielfältige Applikationsprogramme:

- ActiveX-Control (ADS-OCX) für z. B. Visual Basic, Visual C++, Delphi, etc.
- DLL-Interface (ADS-DLL) für z. B. Visual C++ Projekte
- Script-Interface (ADS-Script DLL) für z. B. VBScript, JScript, etc.

System Manager – Eigenschaften

- Verbindung zwischen Server-Prozessabbildern und E/A-Kanälen bitweise
- Standard-Datenformate, z. B. Arrays und Strukturen
- Benutzerdefinierte Datenformate
- fortlaufende Verbindung von Variablen
- Drag und Drop
- Import und Export auf allen Ebenen

Im Folgenden werden das Vorgehen und die Konfigurationsmöglichkeiten im System Manager beschrieben.

Kontextmenü

Abb. 18: Kontextmenü

Box Anfügen... <Einfg>

Fügt CANopen Slaves (Boxen) an. Z.Zt. werden folgende Boxen unterstützt (nähere Beschreibung zu den Boxen folgt weiter hinten):

Unterstützte Boxen	Beschreibung
BK5100 [▶ 29]	Buskoppler
BK5110 [▶ 29]	Economy Buskoppler
BK5120 [▶ 29]	Buskoppler (Nachfolger von BK5100)
LC5100 [▶ 29]	Low-cost Buskoppler
IPxxxx-B510 [▶ 29]	Feldbus Kompakt Box: CANopen Ein/Ausgabebaugruppe in Schutzart IP67
CANopen Node [▶ 31]	Allgemeines CANopen Gerät oder allgemeines CAN-Gerät (Zugriff über CAN-Schicht 2)

Gerät Löschen... <Entf>

Entfernt die FC510x Feldbuskarte und alle untergeordneten Elemente aus der E/A Konfiguration.

Online Reset

Initiiert einen Online Reset auf dem CANopen Bus.

Karteireiter FC510x

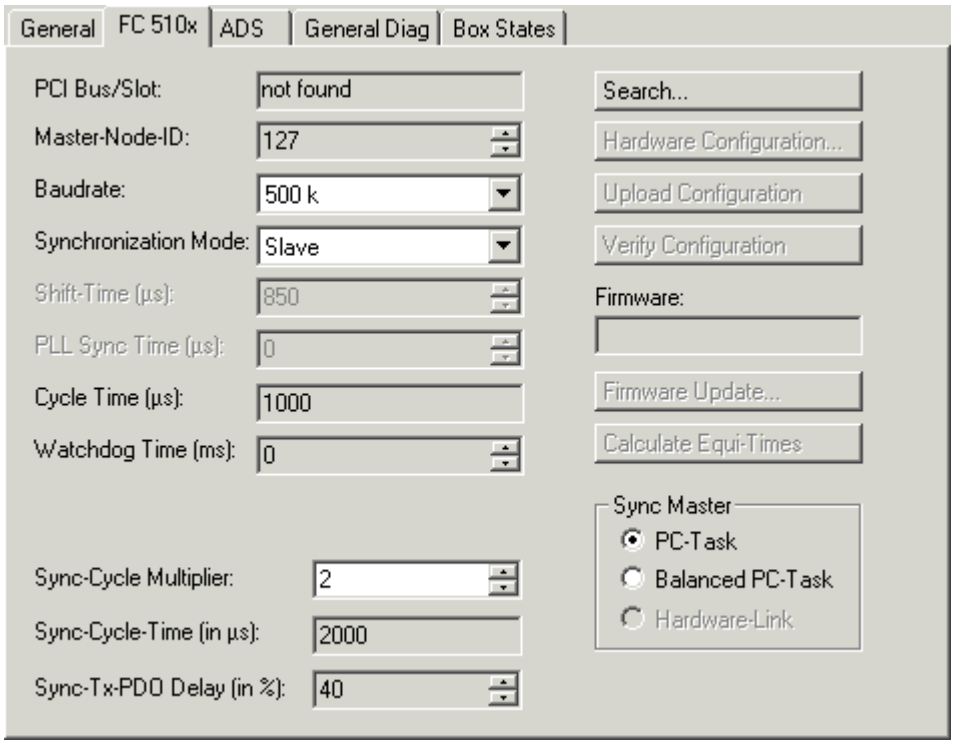


Abb. 19: Karteireiter FC510x

PCI Bus/Slot

Zeigt an in welchem logischen PCI-Slot die Karte gefunden wurde.

Master-Node-ID

Knotenadresse der FC5100. Wertebereich: 1...127. Bestimmt den Identifier des Master-Heartbeat Telegramms. Darf nicht mit einer Slave-Knotenadresse übereinstimmen.

Baudrate

Hier wird die Baudrate [▶ 51] eingestellt. Es wird automatisch überprüft ob die angeschlossenen Slaves diese Baudrate auch unterstützen.

Synchronization Mode

Der Synchronisations-Modus bestimmt die Genauigkeit, mit der das CANopen SYNC [▶ 45] Telegramm generiert wird.

Die höchstpriore Task, die mit der FC510x-Karte verknüpft ist, übernimmt die Ansteuerung der CANopen Karte und ist somit mit dem CANopen Bus synchronisiert. Alle anderen Tasks werden asynchron über entsprechende Puffer bedient. Bei allen Betriebsarten kann die Kommunikationsart für jedes Prozessdatenobjekt (PDO) individuell eingestellt werden - ereignisgesteuert oder synchron (im jeweiligen PDO-Reiter). Falls eines der PDOs für eine synchrone Betriebsart konfiguriert wurde, so wird am Anfang des Zyklus ein SYNC-Telegramm gesendet, mit dem sich die Slaves auf den Master-Zyklus synchronisieren können.

Je nach Anforderung an die SYNC-Genauigkeit können unterschiedliche Modi ausgewählt werden. Hierbei ist zu beachten, dass einzelne SYNC Telegramme bei CAN Systemen Prinzip bedingt um eine Telegrammlänge jittern wenn der Bus zum Zeitpunkt des SYNCs belegt ist. Die SYNC-Genauigkeit bezieht sich also in erster Linie auf die Langzeitstabilität. Busknoten, die sich über ein PLL-Verfahren synchronisieren, sind besonders auf gute Langzeitstabilität angewiesen.

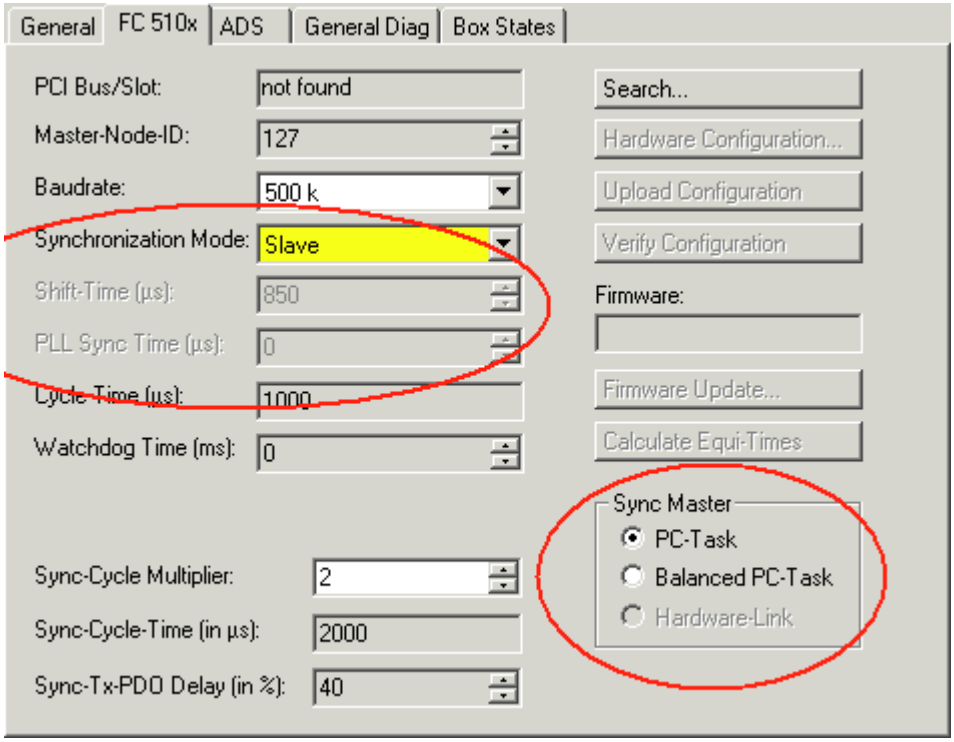


Abb. 20: **Synchronization Mode**

Slave

In der Slave Betriebsart erhält die Karte Ihre Zeitbasis von einem SYNC-Master. Der Sync Master wird über das entsprechende Feld angewählt.

- **Sync Master: PC-Task**. Dies ist die Default Einstellung. Der PC gibt mit Hilfe der TwinCAT-Echtzeit die Zeitbasis vor. Je nach Einstellung löst der Task-Start (Default bei TwinCAT NC) bzw. das Task Ende (Default bei TwinCAT SPS) das SYNC Telegramm aus.
- **Sync Master: Balanced PC Task**. Auch in dieser Betriebsart wird der Sync Zyklus im Mittel mit der Genauigkeit der PC-Zeitbasis erzeugt. Allerdings ist der Abstand zwischen zwei SYNC Telegrammen genauer als beim Sync Master "PC-Task":
  - Laufzeitunterschiede (etwa durch fallabhängige Programmverzweigungen) werden ausgeglichen,
  - die FC510x Karte verzögert anstehende Sende-Telegramme bis nach dem SYNC-Telegramm,
  - die einzelnen SYNC-Abstände werden durch den quarzgenauen Timer der FC510x Karte ermittelt.

Der Karten-Timer wird gegebenenfalls in kleinen Schritten auf den PC-Timer nachjustiert, wenn dieser um den bei "PLL Sync Time" eingestellten Wert vom Karten-Timer abweicht.
Das Sync-Telegramm wird in dieser Betriebsart gegenüber dem Abschluss der TwinCAT Task um die **Shift Time** verzögert. Die Shift Time sollte in dieser Betriebsart möglichst klein gewählt werden - aber ausreichend groß, damit der Prozessdatenzugriff durch die TwinCAT Task erfolgen kann. Bei der Einstellung der optimalen Shift Time hilft die Funktion "Calculate Equi-Times", die nach Erzeugen der Zuordnungen (Mappings) durch Anklicken des entsprechenden Buttons ausgelöst wird.

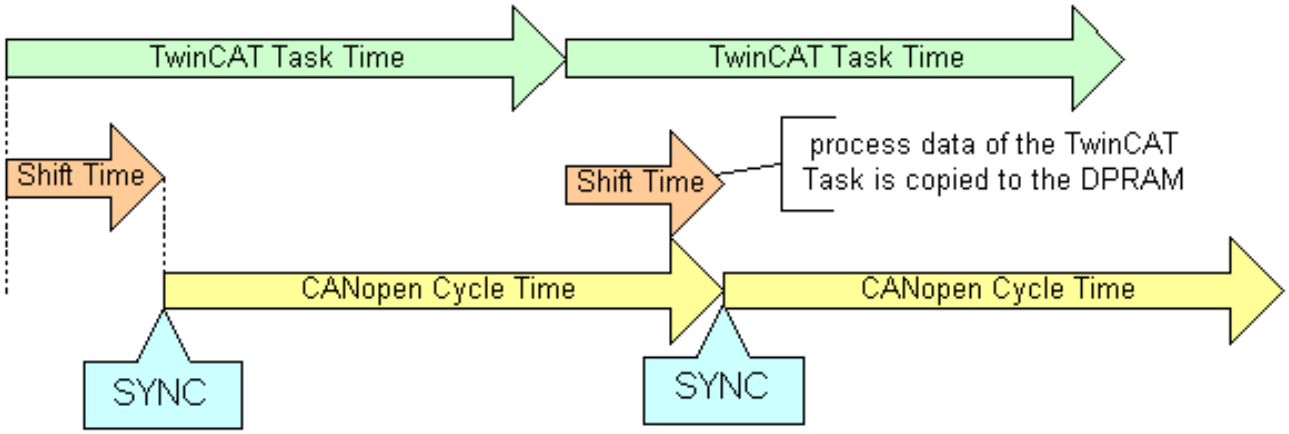


Abb. 21: Synchronization Mode: Slave (Sync Master: Balanced PC Clock)

Master

In der Master Betriebsart erzeugt die Karte Ihre Zeitbasis lokal, das SNYC-Telegramm ist im Mittel quarzgenau. Der Start der TwinCAT Task wird von der Karte vorgegeben und ist gegenüber dem SYNC Telegramm um die Shift Time verzögert. In dieser Betriebsart ist die Shift Time möglichst groß zu wählen. Bei der Einstellung der optimalen Shift Time hilft die Funktion "Calculate Equi-Times", die nach Erzeugen der Zuordnungen (Mappings) durch Anklicken des entsprechenden Buttons ausgelöst wird.

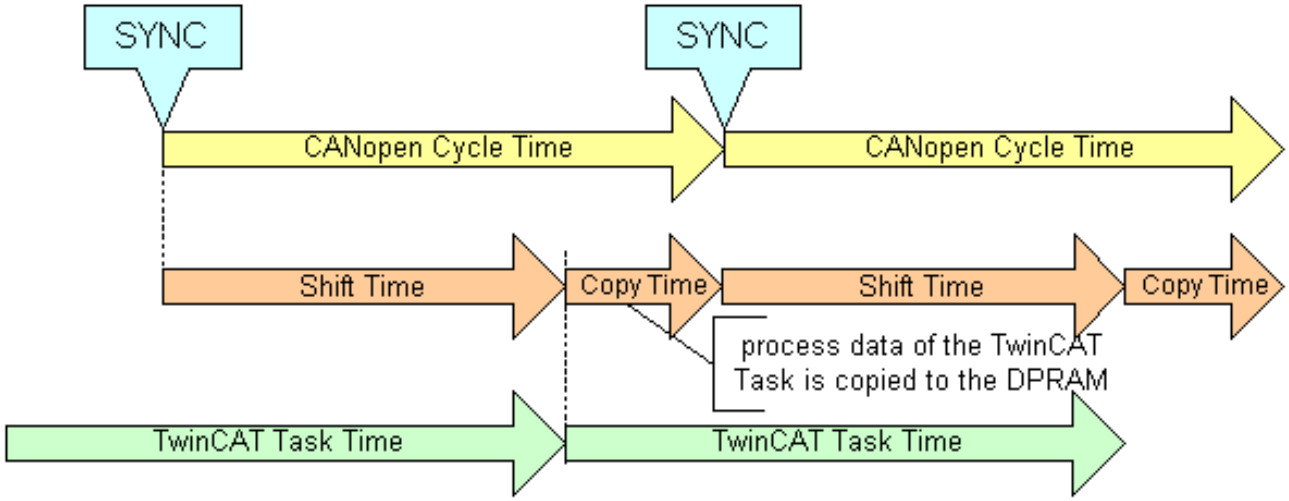


Abb. 22: Synchronization Mode: Master

Cycle-Time

Hier wird die Zykluszeit der zugehörigen höchstprioren Task angezeigt. Die Anzeige wird aktualisiert sobald das Mapping erzeugt wird.

Watchdog Time

Hier kann ein Watchdog aktiviert werden, der dazu führt, dass bei einem Absturz des PCs die FC510x in den STOP-Zustand geht und auch alle projektierten Slaves in diesen Zustand überführt.

Sync-Cycle Multiplier

CANopen SYNC Cycle Time = (Task) Cycle-Time x Sync-Cycle Multiplier. Häufig werden bei CANopen ereignisgesteuerte PDO-Kommunikation und zyklisch synchrone PDO-Kommunikation kombiniert. Um schnell auf ein Ereignis reagieren zu können muss die TwinCAT Task Zyklus-Zeit kleiner sein als die CANopen SYNC Zyklus Zeit. Wird der Sync-Cycle Multiplier auf Werte >1 eingestellt, so wird die TwinCAT Task entsprechend mehrfach aufgerufen bevor das SYNC Telegramm erneut gesendet wird.

Sync-Cycle Time

Hier wird die Zyklus Zeit des CANopen Sync-Telegramms angezeigt. Sie ergibt sich aus der Task-Zykluszeit der höchstprioren Task, deren Prozessdaten mit der Karte verknüpft sind, und aus dem Sync-Cycle Multiplier.

Sync-Tx-PDO Delay

Direkt nach dem SYNC Telegramm senden die synchronisierten Slaves ihre Eingangsdaten bzw. Istwerte. Die FC510x kann das Senden der Ausgangsdaten bzw. Sollwerte (TxPDOs aus Sicht der Karte) verzögern, um den Telegramm-Burst direkt nach dem SYNC zu minimieren. Mit dem Parameter Sync-Tx-PDO Delay wird diese Verzögerung in Prozent der SYNC-Zykluszeit eingestellt.

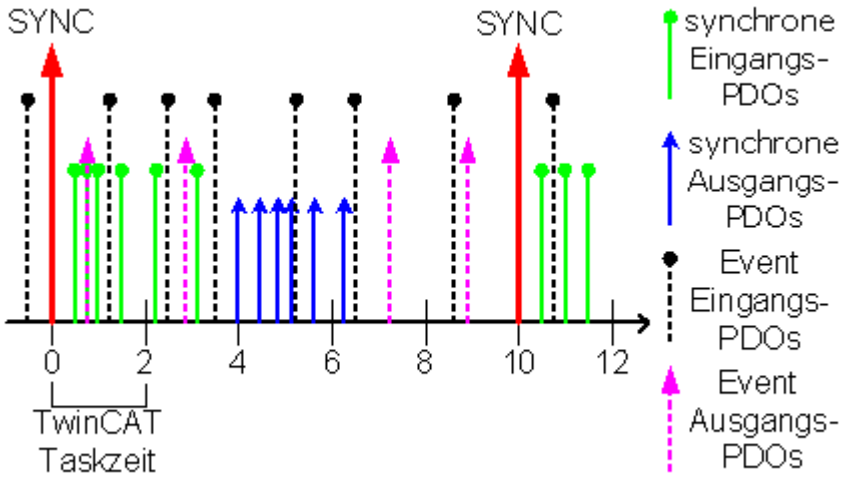


Abb. 23: Beispiel

Task Cycle Time = 2000 µs, Sync-Cycle Multiplier = 5, Sync Tx-PDO Delay =40. Alle 2 ms können ereignisgesteuerte PDOs von der SPS Task verarbeitet werden, der CANopen Sync Cycle beträgt 10 ms, 4 ms (=40 % von 10 ms) nach dem SYNC sendet die FC510x ihre synchronen PDOs.

Search...

Hierüber werden alle vorhandenen FC510x-Kanäle gesucht, und es kann der gewünschte ausgewählt werden. Bei einer FC5102 erscheinen beide Kanäle A und B, die sich logisch wie zwei FC5101-Karten verhalten.

Hardware Configuration...

Hiermit kann die Adresse der FC510x in den unteren Memory-Bereich (unterhalb von 1 MB) des PCs eingestellt werden.

Upload Configuration

Hiermit wird das CANopen Netz gescannt und alle gefunden Geräte werden dem Device (der FC510x) hinzugefügt (es darf keine Box angefügt sein). Bei Beckhoff-Boxen wird die Konfiguration genau ausgelesen, bei Fremdgeräten werden die PDO Konfiguration und das Identity Objekt gelesen und ausgewertet.

Verify Configuration

Erlaubt den Vergleich der erwarteten (eingetragenen) Netzwerk-Konfiguration mit den tatsächlich im Netz vorhandenen Geräten. Es werden die Daten aus dem CANopen Identity Objekt ausgelesen und verglichen. Bei Beckhoff Boxen werden die angeschlossenen Busklemmen bzw. Erweiterungsmodule ausgelesen und verglichen (in Vorbereitung).

Firmware

Hier wird die aktuelle Firmware-Version der FC510x angezeigt.

Firmware Update...

Hierüber kann die Firmware der FC510x-Karte aktualisiert werden. Achtung: Das TwinCAT System muss hierzu gestoppt sein.

Karteireiter ADS

Die FC510x ist ein ADS-Device mit einer eigenen Net-ID, die hier verändert werden kann. Alle ADS-Dienste (Diagnose, azyklische Kommunikation), die an die FC510x gehen, müssen die Karte mittels dieser Net-ID adressieren.

Karteireiter Box States

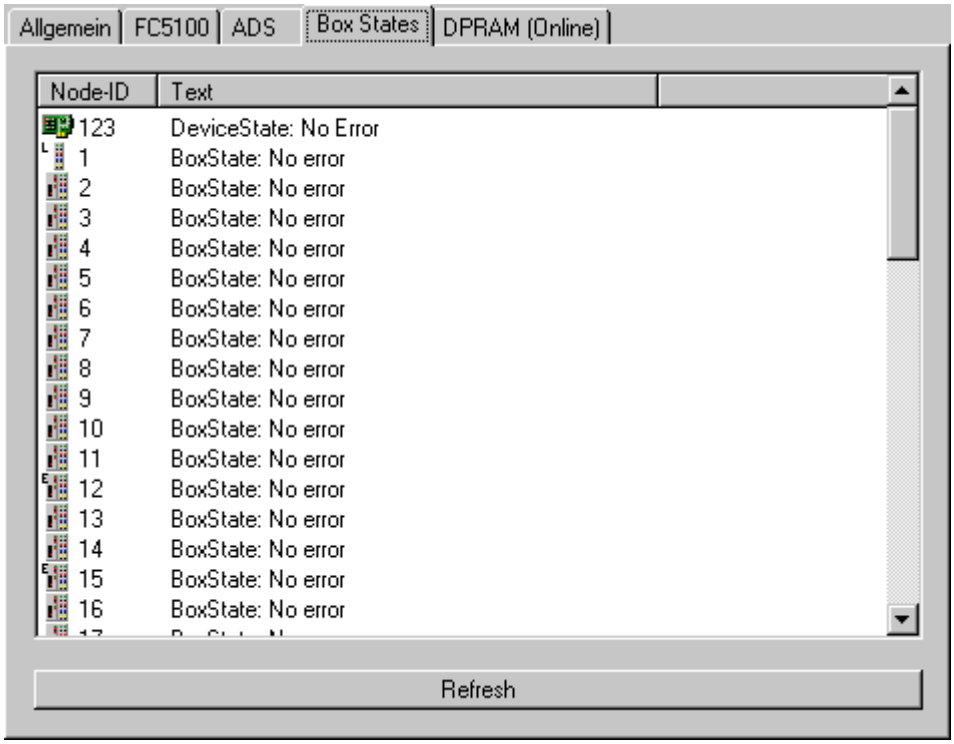


Abb. 24: Karteireiter Box States

Hier wird eine Übersicht aller aktuellen Box-States angezeigt.

Karteireiter DPRAM (Online)

Siehe unter "Online - Anzeige des DPRAMs" in der System Manager Dokumentation.

Diagnose-Eingänge

Die FC510x verfügt automatisch über verschiedene Diagnosevariablen, die den Zustand der Karte und des CANopen-Netzwerks beschreiben:

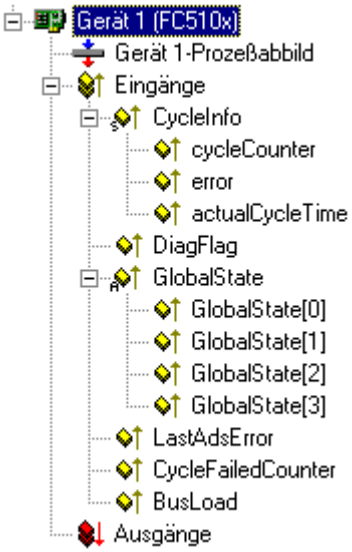


Abb. 25: FC510x - Diagnosevariablen

cycleCounter: wird am Ende jedes Firmware Zyklus inkrementiert, sodass man mit dieser Variable feststellen kann, ob der letzte Zyklus beendet war, bevor die Task gestartet wurde

error: Zeigt die Anzahl der Slaves an, deren Box-State ungleich null ist. Nur wenn dieser Wert ungleich 0 ist, muss der BoxState der Slaves überprüft werden

actualCycleTime: zeigt die aktuelle Zykluszeit in 4/25 µs an. Diese Variable wird nur aktualisiert, falls alle Slaves im Datenaustausch sind (also error gleich 0 ist)

DiagFlag: Zeigt an, ob sich die Diagnoseinformationen der Karte geändert haben, die dann mit ADS-Read ausgelesen werden können. Dazu ist die Net-ID der FC510x, die Port-Nummer 200 und die IndexGroup 0xF100 anzugeben. Der IndexOffset und die Länge beziehen sich dann auf die Diagnosedaten. (Hinweis: die Box States stehen bei den Boxen als Variable auch direkt zur Verfügung.)

Offset 1-127: BusStatus-Liste, je Stationsadresse 1-127 ein Byte, das den Status der Station enthält (s. BoxState bei den CANopen-Boxen)

Global State: Verschiedene Diagnose und Statusanzeigen der FC510x. Das Byte in GlobalState(0) zeigt den Status der Karte in Bezug auf das TwinCAT System an: RUN, RESET, OFFLINE und STOP werden unterschieden. GlobalState(2) gibt Informationen über den Zustand des CAN Controllers: "CAN Warning Limit erreicht" und "Bus-Off" werden angezeigt. Warning Limit erreicht bedeutet dass der Sende- oder Empfangsfehlerzähler des CAN Controllers den Wert 96 überstiegen hat; BusOff bedeutet dass der CAN Controller aufgrund zu vieler CAN Fehler (Error Frames) nicht mehr am Busverkehr teilnimmt. In diesem Fall liegt ein schwerwiegender physikalischer Fehler im CAN-Netz vor. (z. B. zu wenige oder zu viele Abschlusswiderstände, mindestens ein Teilnehmer mit falscher Baudrate, Kurzschluss, etc.) Der Zustand Bus-Off kann nur durch einen Reset der Karte verlassen werden. Details über weitere Global State Daten siehe Kommentar im "Online"-Reiter.

LastAdsError: zeigt den Fehlercode des letzten aufgetretenen ADS-Zugriffsfehlers an - z. B. wenn versucht wurde, die Diagnosedaten eines deaktivierten Knotens zu lesen.

CycleFailedCounter: zählt die Anzahl der Firmware Zyklen, die nicht beendet werden konnten, bevor die zugehörige Task wieder das Prozessabbild lesen bzw. schreiben wollte. Falls dieser Zähler inkrementiert wird, so ist die Taskzykluszeit zu klein gewählt für die aktuelle Netzwerkkonfiguration.

BusLoad: zeigt die aktuelle Busbelastung in % an. Die Buslast [▶ 51] ist ein wichtiges Auslegungskriterium für CAN Netzwerke. Dargestellt wird ein Mittelwert über 100 ms.

4.2 Beckhoff Buskoppler

Der Buskoppler BK51x0 sowie die FeldbusBox IPxxx-B510 werden im **CANopen** Bus eingesetzt. Nachfolgend werden die spezifischen Eigenschaften beschrieben, die sich von anderen Buskopplern bzw. Feldbus Box Modulen unterscheiden.

Typen	Beschreibung
BK5100	Buskoppler
BK5110	Economy Buskoppler
BK5120	Buskoppler, Nachfolger von BK5100
LC5100	Low-cost Buskoppler
IPxxxx-B510	Feldbus Kompakt Box: CANopen Ein-/Ausgabebaugruppe in Schutzart IP67
ILxxxx-B510	Feldbus Koppler Box: Erweiterbare CANopen Ein-/Ausgabebaugruppe in Schutzart IP67

Karteireiter BK51x0/IX-B510

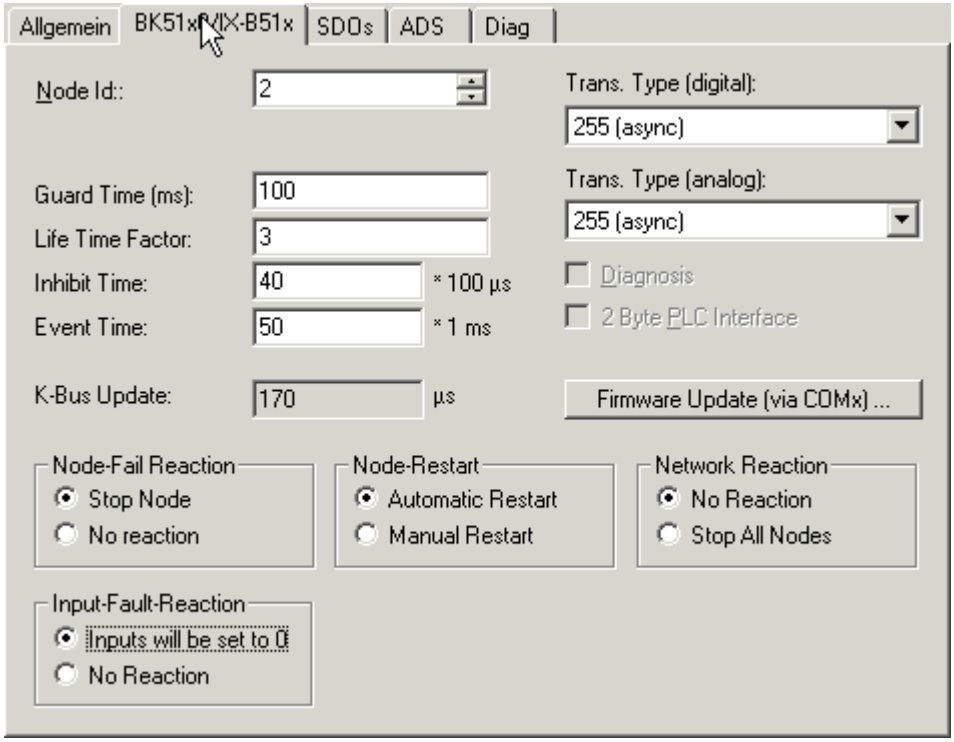


Abb. 26: Karteireiter BK51x0/IX-B510

Node Id: Stellt die Node ID des CAN Bus Teilnehmers ein (zwischen 1 und 63 (BK51x0) bzw. 1 und 99 (IPxxxx-B510)). Dieser Wert muss mit dem am Buskoppler bzw. an der Kompakt Box eingestellten Wert übereinstimmen.

Guard Time: Zykluszeit für die Knotenüberwachung (Nodeguarding).

Life Time Factor: Mit Guard Time multipliziert ergibt sich die Watchdog-Zeit für die Überwachung des Masters durch den Koppler (Lifeguarding). Lifeguarding ist deaktiviert, wenn Life Time Factor zu null gesetzt wird.

Inhibit Time: Gibt den minimalen Sendeabstand für PDOs (Telegramme) mit analogen und Sondersignalen an. Wenn mehr als 64 digitale Signale vorhanden sind, werden diese auch mit dieser Inhibit-Zeit [▶ 49] versehen.

Event Time: Ereignis-Timer für Sende-PDOs. Der Ablauf dieses Timers wird als zusätzlich eingetretenes Ereignis für das entsprechende PDO gewertet, das PDO wird also dann gesendet. Wenn das Applikationsereignis während einer Timer-Periode auftritt, so wird ebenfalls gesendet und der Timer wird zurückgesetzt.

K-Bus Update: Berechnet die voraussichtliche Dauer für einen vollständigen Update des Klemmenbusses (ist abhängig von Anzahl und Art der angeschlossenen Klemmen).

Trans.Type: Gibt den Transmission Type [▶ 45] für digitale bzw. analoge Eingangstelegramme an. 254 + 255 entspricht der Ereignisgesteuerten Übertragung, 1...240 sind synchrone Übertragungsarten. Näheres siehe auch Handbuch BK51X0.

Firmware Update: Ermöglicht die Aktualisierung der Koppler-Firmware über die serielle Schnittstelle (erfordert Schnittstellenkabel des KS2000 Softwarepakets).

Karteireiter SDOs

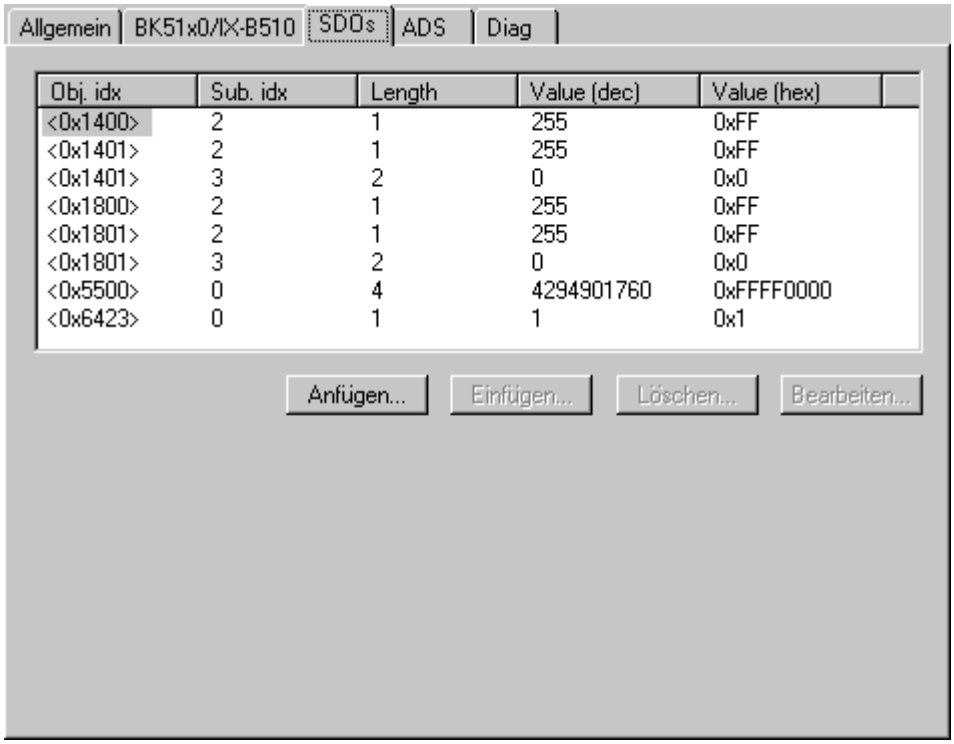


Abb. 27: Karteireiter SDOs

Auf dieser Seite werden SDO Einträge angezeigt/verwaltet, die beim Startup zum Knoten geschickt werden. Einträge deren Objekt-Index in spitzen Klammern stehen, sind automatisch aufgrund der aktuellen Klemmenkonfiguration erzeugt worden. Weitere Einträge können über "Anfügen", "Einfügen", "Löschen" und "Bearbeiten" verwaltet werden.

Karteireiter ADS

Um SDO-Objekte auch zur Laufzeit schreiben und lesen zu können (z. B. aus der SPS heraus), wird dem Knoten (Buskoppler) ein ADS-Port zugewiesen. Dieser kann bei Bedarf verändert werden. Die ADS IndexGroup beinhaltet den CANopen Object Index und der ADS IndexOffset beinhaltet den CANopen SubIndex. Details zur SDO Kommunikation via ADS siehe Kapitel SDO Kommunikation [▶ 57].

Karteireiter Diag

Hier werden Diagnoseinformationen dargestellt. Der Fensterinhalt wird nicht zyklisch aufgefrischt, bei Bedarf den "Refresh" Button anwählen. Die dargestellten Diagnose-Informationen können auch per ADS abgefragt werden [▶ 63].

4.3 CANopen Knoten

CANopen Geräte, die nicht im TwinCAT System Manager bekannt sind, können durch Anwahl der Box "CANopen Node" ins Netz aufgenommen werden. Für diese Geräte können die CAN(open)-Nachrichten (PDOs) direkt konfiguriert werden. Damit wird die maximale Flexibilität dieser allgemeinen CANopen Schnittstelle gewährleistet.

Bei Verwendung der FC510x / EL6751 können mit Hilfe dieser Box auch beliebige CAN Identifier gesendet und empfangen werden - damit ist die Kommunikation mit beliebigen CAN Knoten möglich. Einzige Voraussetzung ist die Unterstützung mindestens einer der von der FC510x / EL6751 unterstützten Baudraten [▶ 61].

Karteireiter CAN Node

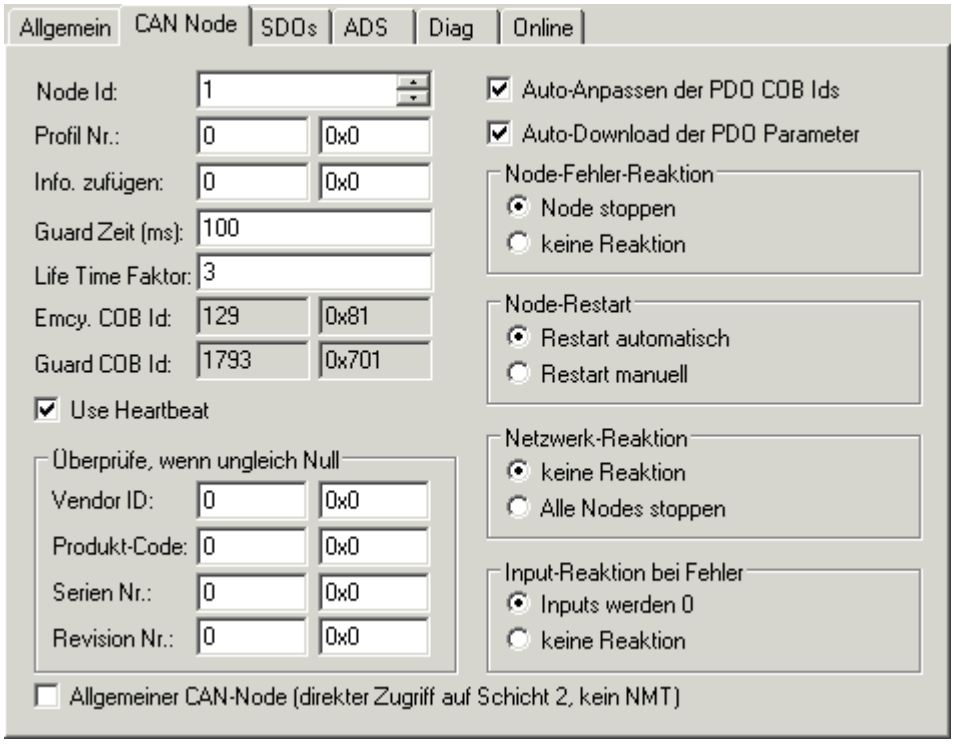


Abb. 28: Karteireiter CAN Node

Node ID
Hier wird die Knotenadresse des allgemeinen CANopen Gerätes eingestellt. Wenn die Box "Auto Anpassen der PDO COB Ids" angewählt ist, so werden die Default-Identifier der Prozessdatenobjekte bei Änderung der Node-ID entsprechend nachgeführt.

Profil Nr.
Nach CANopen enthält der Parameter 0x1000 "Device Type" in den beiden niederwertigsten Bytes die Nummer des vom Gerät unterstützten Geräteprofils. Diese wird hier eingetragen und beim Systemstart mit dem im Gerät vorhandenen Parameter verglichen. Falls kein Geräteprofil unterstützt wird, so enthält der Parameter den Wert 0.

Info zufügen
Die Additional Info steht in den beiden höchstwertigen Bytes des Objektverzeichniseintrages 0x1000 (Device Type).

Der Vergleich Soll-/ Ist-Konfiguration erfolgt nur, wenn Profile No. oder Add. Info (also Objektverzeichniseintrag 0x1000) auf Wert ungleich null konfiguriert sind. Falls die erwarteten Werte beim Systemstart nicht mit den vorhandenen übereinstimmen, so wird der Start dieses Knotens abgebrochen und eine entsprechende Fehlermeldung im Diag-Reiter angezeigt.

Guard Zeit
Die Guard Time bestimmt das Intervall, in dem der Knoten überwacht wird (Node Guarding). 0 bedeutet keine Überwachung. Der eingetragene Wert wird auf das nächste Vielfache von 10 ms aufgerundet.

Life Time Faktor
Guard Time x Life Time Factor bestimmt die Watchdog-Länge für die gegenseitige Überwachung von Karte und CANopen Knoten. 0 bedeutet, dass der CANopen Knoten die Karte nicht überwacht. Bei 0 nimmt die Karte die Guard Time direkt als Watchdog-Länge.

Die FC 510x / EL6751unterstützen auch das Heartbeat-Protokoll und versuchen zunächst diese Form der Knotenüberwachung auf dem CANopen-Knoten zu starten (Schreibzugriff auf die Objekte 0x1016 und 0x1017 im Objektverzeichnis). Falls dieser Versuch fehlschlägt, wird Guarding aktiviert. Eingetragen werden die Guard Time als Producer Heartbeat Time und (Guard Time x Life Time Factor) als Consumer Heartbeat Time. In diesem Fall wird ein Heartbeat Telegramm mit der kleinsten konfigurierten Guard Time gesendet (die Guard Times können für jeden Knoten individuell eingestellt werden).

Emcy COB Id / Guard COB ID
Identifier für Emergency Nachrichten bzw. Guarding Protocol. Diese ergeben sich aus der Knotenadresse.

Use Heartbeat
Zur Überwachung des Knoten wird Heartbeat verwendet. Ist dies deaktiviert wird das Guarding zur Überwachung verwendet.

Auto-Anpassen PDO...
Gibt an, ob TwinCAT die PDO-Kommunikationsparameter beim Systemstart zum Knoten downloaden soll.

Falls der Download der PDO Parameter Identifier und Transmission Type fehlschlägt versucht die Karte diese Parameter zu lesen und mit den konfigurierten Werten zu vergleichen. Auf diese Weise werden auch Knoten unterstützt, die z. B. die Default-Identifier als read-only Werte implementiert haben.

Vendor ID, Product Code, Serial Nr., Revision Nr.
Falls hier Werte ungleich null eingetragen sind, so werden diese Einträge des Identity Objektes (0x1018 im Objektverzeichnis) beim Systemstart ausgelesen und mit den konfigurierten Werten verglichen. Nur wenn die Werte übereinstimmen, wird der entsprechende Knoten gestartet. Es ist auch möglich, nur einen Teil der Werte (z. B. die Vendor ID und den Product Code) zu vergleichen - dann sind lediglich die nicht gewünschten Parameter auf null zu setzen.

Node-Fehler Reaktion

Stop Node
Nach einem erkannten Knotenfehler wird der Knoten in den "Stopped" Zustand gesetzt (NMT Kommando "Stop Remote Node"). Damit können die Knoten (je nach Geräteprofil) über die Netzwerkstatusmaschine in einen sicheren Zustand gezwungen werden - ein Ansprechen über SDO ist dann allerdings nicht mehr möglich.

Keine Reaktion
Kein NMT Stop Remote Node Kommando nach Knotenfehler

Node-Restart

Restart automatisch
Nach einem erkannten Knotenfehler versucht die Karte automatisch, den Knoten wieder aufzustarten. Der Aufstartversuch wird von einem Reset-Node Kommando eingeleitet.

Restart manuell
Nach Knotenfehler bleibt dieser Knoten im Fehlerzustand und wird nicht automatisch gestartet. Ein Neustart ist über "I/O-Reset" möglich.

Netzwerk Reaktion

Keine Reaktion
Der Ausfall eines Knotens hat keine Auswirkungen auf die anderen Busteilnehmer

Alle Nodes stoppen
Nach Ausfall eines Knotens werden alle anderen zuvor gestarteten Knoten gestoppt (NMT Kommando stop remote node). Ein Neustart des Systems ist dann erforderlich.

Allgemeiner CAN Node
Wenn diese Checkbox angewählt ist, so ist das gesamte CANopen Netzwerkmanagement für diesen Teilnehmer deaktiviert: er wird nicht gestartet, überwacht etc.. Die PDO-Einträge werden als reine CAN (Schicht 2-) Telegramme aufgefasst und ereignisgesteuert der Steuerung zur Verfügung gestellt.

CANopen Terminologie

Da die CANopen Terminologie auch beim allgemeinen CAN Knoten beibehalten wird, ist zu berücksichtigen, dass RxPDOs die Telegramme sind, die von der FC510x / EL6751 gesendet werden und TxPDOs die empfangenen Telegramme sind.

Mit dieser Option lassen sich beliebige CAN Knoten an TwinCAT anbinden, falls die Baudrate [▶ 61] und die Bit-Timing Parameter übereinstimmen. Das jeweilige Protokoll kann dann im SPS Programm nachgebildet werden. Es ist auch möglich, CANopen-Teilnehmer und allgemeine CAN Knoten im gleichen Netz zu betreiben - falls es keine Identifier-Überschneidungen gibt (Systembedingt können keine Identifier doppelt verwendet werden).

CANopen PDOs

Prozessdatenobjekte [▶ 45] (PDOs) sind CAN-Telegramme, die Prozessdaten ohne Protokoll-Overhead transportieren. RxPDOs werden vom Knoten empfangen, TxPDOs werden vom Knoten gesendet. Diese Bezeichnung wird im System Manager aus Sicht des konfigurierten Knotens beibehalten, d.h. RxPDOs werden von TwinCAT gesendet, TxPDOs werden von TwinCAT empfangen.

Karteireiter PDO

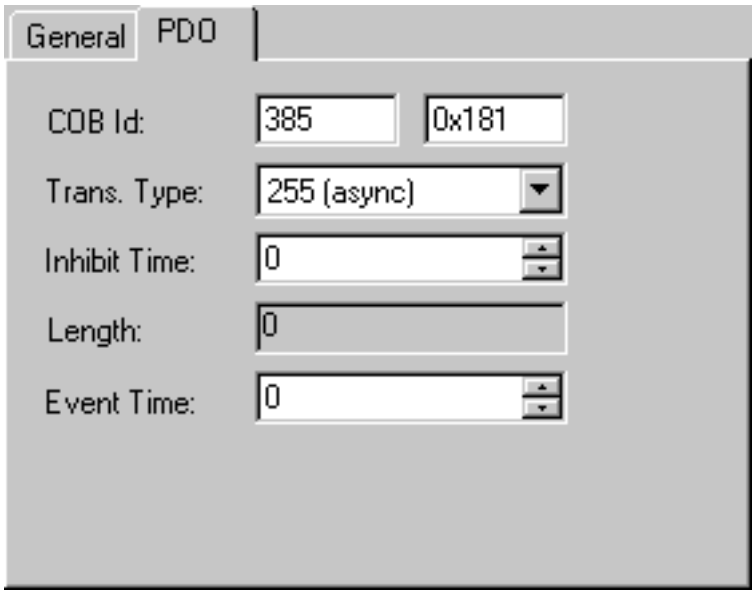


Abb. 29: Karteireiter PDO

COB Id
Der CAN-Identifier dieses PDOs. Für jeweils zwei Sende- und Empfangs-PDOs je Knoten stellt CANopen Default-Identifier [▶ 61] zur Verfügung. Diese können dann geändert werden.

Trans.Typ
Der Transmission Type [▶ 45] bestimmt das Sendeverhalten des PDOs. 255 entspricht dem ereignisgesteuerten Senden.

Inhibit Time
Sendeverzögerung [▶ 45] zwischen zwei gleichen PDOs. Wird in Vielfachen von 0,1 ms angegeben.

Länge
Die Länge des PDOs ergibt sich aus den gemappten Variablen und kann daher hier nicht editiert werden.

Event Time (nur FC510x und EL6751)
Hier wird der Wert für den Event Timer [▶ 49] in ms eingetragen. Bei Sende-PDOs (hier: RxPDOs, siehe oben) löst der Ablauf dieses Timers ein zusätzliches Senden des PDOs aus, bei Empfangs-PDOs (hier: TxPDOs) wird das Eintreffen eines PDOs innerhalb des eingestellten Wertes überwacht und ggf. der Box-State des Knotens verändert. Bei 0 wird der Parameter nicht zum Knoten übertragen.

TwinCAT erzeugt aus den hier eingegebenen Parametern entsprechende Einträge im Objektverzeichnis des Knotens, die beim Systemstart über SDO übertragen werden. Die Einträge können beim Karteireiter SDOs eingesehen werden. Ist dieses Verhalten unerwünscht, so kann es über die Checkbox "Auto-Download der PDO Parameter" beim Karteireiter CAN Node deaktiviert werden.

Prüfung der PDO-Größe deaktivieren
Checkbox zur Deaktivierung der Längenüberprüfung der PDO-Größe.

Baumdarstellung

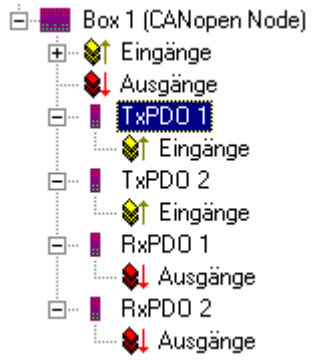


Abb. 30: Baumdarstellung

TwinCAT sieht für einen allgemeinen CANopen-Knoten zunächst je zwei Sende- und Empfangs-PDOs vor, die mit Default-Identifiern [▶ 61] versehen sind. Überzählige PDOs können angewählt und entfernt werden.

TxPDOs werden vom CANopen Knoten gesendet und enthalten im allgemeinen Fall Eingänge. RxPDOs werden vom Knoten empfangen, also von TwinCAT gesendet

Die PDOs werden mit Variablen gefüllt, indem man "Eingänge" bzw. "Ausgänge" mit der rechten Maustaste anklickt und die entsprechende(n) Variablen auswählt. Werden mehrere Variablen des gleichen Typs mit einer Aktion eingefügt, so wird der Offset innerhalb des PDOs automatisch erzeugt. Werden Variablen hintereinander eingefügt, so ist der entsprechende Offset (Start-Adresse innerhalb des CAN-Telegramms) für jede Variable einzustellen.

Objektverzeichniseinträge in TwinCAT

TwinCAT ordnet die PDOs der angezeigten Reihenfolge nach den Objektverzeichniseinträgen im Knoten zu. So werden z. B. die PDO Kommunikationsparameter des dritten aufgelisteten TxPDOs stets auf Index 0x1802 geschrieben - unabhängig von der Bezeichnung des PDOs im System Manager. Falls also nur PDO1 und PDO3 verwendet werden sollen, so ist ein PDO2 ebenfalls einzutragen - in diesem Fall ohne das Variablen zugeordnet werden. PDOs ohne Variablen werden nicht gesendet und auch nicht erwartet.

Kontextmenü

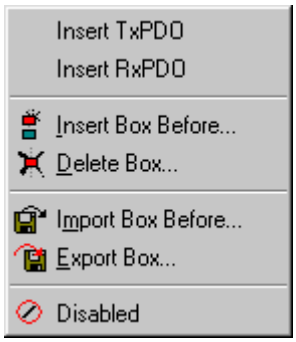


Abb. 31: Kontextmenü

Das obenstehende Menü erhält man, indem man den allgemeinen CANopen Knoten mit der rechten Maustaste anklickt. Hier können weitere Tx- bzw. Rx-PDOs eingefügt werden.

Karteireiter SDOs

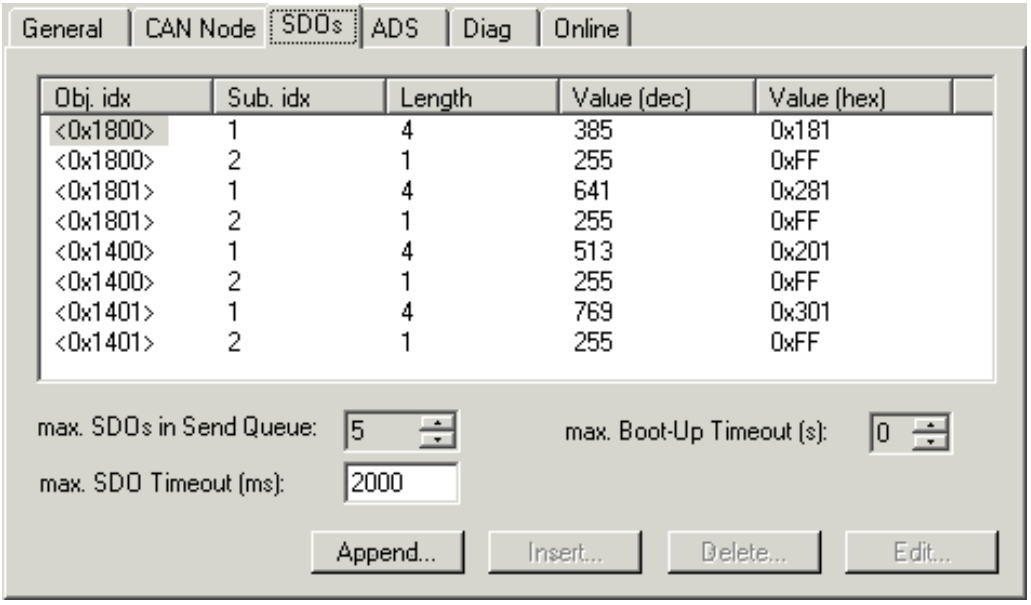


Abb. 32: Karteireiter SDOs

Auf dieser Seite werden SDO Einträge angezeigt/verwaltet, die beim Startup zum Knoten geschickt werden. Einträge deren Objekt-Index in spitzen Klammern stehen, sind automatisch aufgrund der aktuellen Klemmenkonfiguration erzeugt worden. Weitere Einträge können über "Anfügen", "Einfügen", Löschen" und "Bearbeiten" verwaltet werden.

Karteireiter ADS

Um SDO-Objekte auch zur Laufzeit schreiben und lesen zu können (z. B. aus der SPS heraus), kann dem Knoten (Buskoppler) ein ADS-Port zugewiesen werden (CIFx0-CAN). Die FC510x / EL6751 verfügt stets über einen ADS-Port für jeden Knoten, da die Diagnoseinformationen über ADS transportiert werden. Über diesen können SDO-Objekte per ADS Read Request bzw. Write Request gelesen und geschrieben werden.

Der ADS IndexGroup beinhaltet den CANopen Object Index und der ADS IndexOffset beinhaltet den CANopen SubIndex.

4.4 Konfigurationsdateien

eds Files (electronic data sheets) stellen die Konfigurationsmöglichkeiten eines CANopen Gerätes in einem standardisierten Datenformat zur Verfügung. Das Datenformat ist in CiA DSP 306 festgelegt. Ein Tool zur Überprüfung dieses Datenformats steht beim Nutzerverband CAN-in-Automation e.V. zur Verfügung.

eds Files für die Beckhoff CANopen Slave Geräte stehen auf der Beckhoff Website zum Download zur Verfügung.

Nachdem sich viele eds-Files nicht vollständig an den Standard halten, hat Beckhoff bislang auf die Unterstützung von eds Files im System Manager verzichtet. Das direkte Konfigurieren von PDO Parametern erlaubt es, sich direkt an die einzubindenden Geräte anzupassen und auch Geräte einzubinden, die nicht vollständig dem Standard entsprechen.

5 CANopen Kommunikation

5.1 Netzwerkmanagement

Einfacher Boot-Up

CANopen erlaubt einen sehr einfachen Boot-Up des verteilten Netzwerkes. Die Module befinden sich nach der Initialisierung automatisch im Zustand *Pre-Operational*. In diesem Zustand kann bereits über Service-Datenobjekte (SDOs) mit Default-Identifiern auf das Objektverzeichnis zugegriffen werden, die Module können also konfiguriert werden. Da für alle Einträge im Objektverzeichnis Default-Einstellungen vorhanden sind, kann in den meisten Fällen auf eine Konfiguration verzichtet werden.

Zum Starten der Module ist dann nur eine einzige CAN-Nachricht erforderlich: Start_Remote_Node: Identifier *0*, zwei Datenbytes: 0x01, 0x00. Sie überführt die Knoten in den Zustand *Operational*.

Netzwerkstatus

Die Zustände im CANopen Boot-Up und die Zustandsübergänge sind aus dem Zustandsdiagramm ersichtlich:

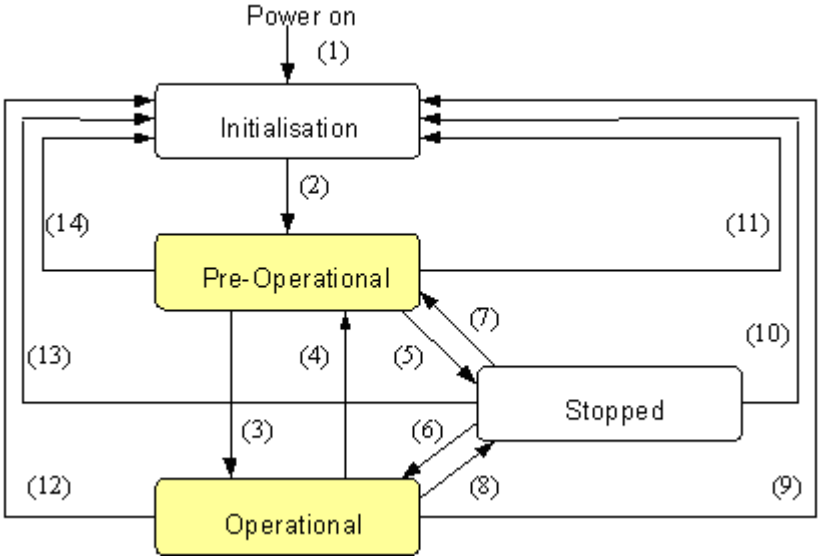


Abb. 33: Zustandsdiagramm CANopen Boot-up

Pre-Operational

Nach der Initialisierung geht der Buskoppler automatisch, d.h. ohne Befehl von außen, in den Zustand *Pre-Operational* über. In diesem Zustand kann er konfiguriert werden, denn die Servicedatenobjekte (SDOs) sind bereits aktiv. Die Prozessdatenobjekte sind hingegen noch gesperrt.

Operational

Im Zustand *Operational* sind auch die Prozessdatenobjekte aktiv.

Wenn der Buskoppler aufgrund äußerer Einflüsse (z. B. CAN-Störung, keine Ausgangs-Spannung) oder innerer Einflüsse (z. B. K-Bus-Fehler) nicht mehr in der Lage ist, Ausgänge zu setzen oder Eingänge zu lesen bzw. zu kommunizieren, so versucht er eine entsprechende Emergency-Nachricht zu senden, geht in den Fehlerzustand und fällt dabei in den Zustand *Pre-Operational* zurück. Damit kann auch die NMT-Statusmaschine des Netzwerkmasters fatale Fehler sofort erkennen.

Stopped

Im Zustand *Stopped* (früher *Prepared*) ist keine Datenkommunikation mit dem Koppler möglich - lediglich NMT-Nachrichten werden empfangen. Die Ausgänge gehen in den Fehlerzustand.

Statusübergänge

Die Netzwerkmanagement-Nachrichten haben einen sehr einfachen Aufbau: CAN-Identifier *0* mit zwei Byte Dateninhalt. Das erste Datenbyte enthält den sogenannten Command-Specifier (cs), das zweite Datenbyte die Knotenadresse, wobei die Knotenadresse *0* alle Knoten anspricht (Broadcast).

11-bit Identifier	2 Byte Nutzdaten							
0x00	cs	Node-ID						

Die folgende Tabelle gibt einen Überblick über alle CANopen Statusübergänge und die dazugehörigen Kommandos (Command Specifier im NMT Master-Telegramm):

Statusübergang	Command Specifier cs	Erläuterung
(1)	-	Der Initialisierungs-Status wird beim Einschalten selbsttätig erreicht
(2)	-	Nach der Initialisierung wird der Status Pre-Operational automatisch erreicht - dabei wird die Boot-Up-Nachricht abgeschickt.
(3), (6)	cs = 1 = 0x01	Start_Remote_Node. Startet Modul, gibt Ausgänge frei, Startet Übertragung von PDOs.
(4), (7)	cs = 128 = 0x80	Enter_Pre-Operational. Stoppt PDO-Übertragung, SDO weiter aktiv.
(5), (8)	cs = 2 = 0x02	Stop_Remote_Node. Ausgänge gehen in den Fehlerzustand, SDO und PDO abgeschaltet.
(9), (10), (11)	cs = 129 = 0x81	Reset_Node. Führt Reset durch. Alle Objekte werden auf Power-On Defaults zurückgesetzt.
(12), (13), (14)	cs = 130 = 0x82	Reset_Communication. Führt Reset der Kommunikationsfunktionen durch. Objekte 0x1000 - 0x1FFF werden auf Power-On Defaults zurückgesetzt

Beispiel 1

Mit folgendem Telegramm werden netzwerkweit alle Baugruppen in den Fehlerzustand (Ausgänge sicherer Zustand) überführt:

11-bit Identifier	2 Byte Nutzdaten							
0x00	0x02	0x00						

Beispiel 2

Mit folgendem Telegramm wird Knoten 17 zurückgesetzt (resetted):

11-bit Identifier	2 Byte Nutzdaten							
0x00	0x81	0x11						

Boot-Up-Nachricht

Nach der Initialisierungsphase und dem Selbsttest sendet der Buskoppler die Boot-Up-Nachricht, eine CAN-Nachricht mit einem Datenbyte (0) auf dem Identifier der Guarding- bzw. Heartbeat-Nachricht: CAN-ID = 0x700 + Node-ID. Damit kann ein temporärer Ausfall einer Baugruppe während des Betriebs (z. B. durch einen Spannungseinbruch) oder eine nachträglich eingeschaltete Baugruppe zuverlässig auch ohne Node Guarding festgestellt werden. Der Sender kann über den Identifier der Nachricht (siehe Default-Identifier-Verteilung) bestimmt werden.

Außerdem ist es mit Hilfe der Boot-Up-Nachricht möglich, die beim Aufstarten am Netz befindlichen Knoten mit einem einfachen CAN-Monitor zu erkennen, ohne dass ein Schreibzugriff (z. B. Scannen des Netzwerks durch Auslesen von Parameter 0x1000) auf den Bus erforderlich ist.

Schließlich wird durch die Boot-Up-Nachricht das Ende der Initialisierungsphase kommuniziert; der Buskoppler signalisiert, dass er nun konfiguriert bzw. gestartet werden kann.

Firmwarestand BA

Bis Firmwarestand BA wurde für die Boot-Up-Nachricht der Emergency Identifier genutzt.

Format Boot-Up Nachricht

11-bit Identifier	1 Byte Nutzdaten							
0x700 (=1792) + Node-ID	0x00							

Knotenüberwachung

Für die Ausfallüberwachung des CANopen Netzwerkes stehen Heartbeat und Guarding-Mechanismen zur Verfügung. Diese sind bei CANopen besonders wichtig, da sich die Baugruppen in der ereignisgesteuerten Betriebsart nicht regelmäßig melden. Beim Guarding werden die Teilnehmer per Datenanforderungstelegramm (Remote Frame) zyklisch nach ihrem Status gefragt, beim Heartbeat senden die Knoten ihren Status von selbst.

Guarding: Node Guarding und Life Guarding

Über Node Guarding werden die dezentralen Peripherie-Baugruppen überwacht, die ihrerseits über Life Guarding den Ausfall des Guarding-Masters erkennen können. Beim Guarding setzt der Master Remote Frames (remote transmit request, Nachrichten-Anforderungstelegramme) auf die Guarding Identifier der zu überwachenden Slaves ab. Diese antworten mit der Guarding-Nachricht. Diese enthält den Status-Code des Slaves sowie ein Toggle-Bit, das nach jeder Nachricht wechseln muss. Falls Status- oder Toggle-Bit nicht mit den vom NMT-Master erwarteten übereinstimmen oder falls keine Antwort erfolgt geht der Master von einem Slave-Fehler aus.

Guarding-Verfahren

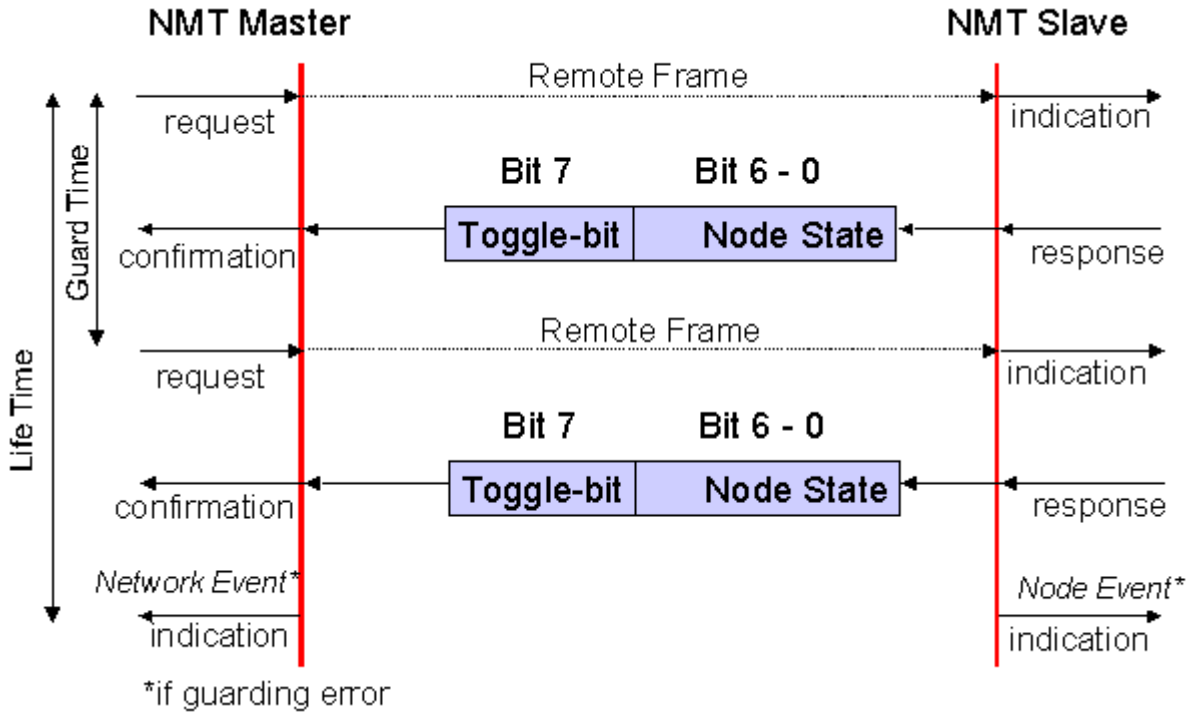


Abb. 34: Schematische Darstellung „Guarding-Verfahren“

Protokoll

Das im ersten Guarding-Telegramm übertragene Toggle-Bit (t) hat den Wert *0*. Anschließend wechselt (toggelt) das Bit in jedem Guarding-Telegramm und signalisiert so, ob ein Telegramm verloren ging. In den restlichen sieben Bit gibt der Knoten seinen Netzwerk Status (s) an:

s	Status
4 = 0x04	Stopped (früher: Prepared)
5 = 0x05	Operational
127 = 0x7F	Pre-Operational

Beispiel

Die Garding Nachricht des Knotens 27 (0x1B) muss mit einem Remote Frame mit Identifier 0x71B (1819_{dez}) angefragt werden. Wenn der Knoten *Operational* ist, wechselt das erste Datenbyte der Antwort-Nachricht zwischen 0x05 und 0x85, im Zustand *Pre-Operational* wechselt es zwischen 0x7F und 0xFF.

Guard Time und Life Time Factor

Wenn der Master die Guard-Nachrichten streng zyklisch anfordert, kann der Slave den Ausfall des Masters erkennen. Falls der Slave in diesem Fall innerhalb der eingestellten *Node Life Time* keine Nachrichtenanforderung vom Master erhält (Guarding-Fehler), geht er von einem Masterausfall aus (Watchdog-Funktion). Dann setzt er seine Ausgänge in den Fehlerzustand, sendet ein Emergency-Telegramm und fällt in den Zustand Pre-Operational zurück. Nach einem Guarding Time-Out kann das Verfahren durch Übertragen eines erneuten Guarding-Telegramms wieder angeregt werden.

Die Node Life-Time berechnet sich aus den Parametern Guard-Time (Objekt 0x100C) und Life-Time-Factor (Objekt 0x100D):

Life-Time = Guard-Time x Life-Time-Factor

Falls einer der beiden Parameter "0" ist (Default-Einstellung), erfolgt keine Überwachung des Masters (kein Life Guarding).

Heartbeat: Knotenüberwachung ohne Remote Frame

Beim Heartbeat-Verfahren senden die Knoten ihre jeweilige Statusmeldung zyklisch selbsttätig. Es kann daher auf Remote Frames verzichtet werden und es wird weniger Buslast erzeugt als beim Guarding-Verfahren.

Der Master sendet sein Heartbeat-Telegramm ebenfalls zyklisch, die Slaves können somit den Ausfall des Masters ebenfalls erkennen.

Heartbeat-Verfahren

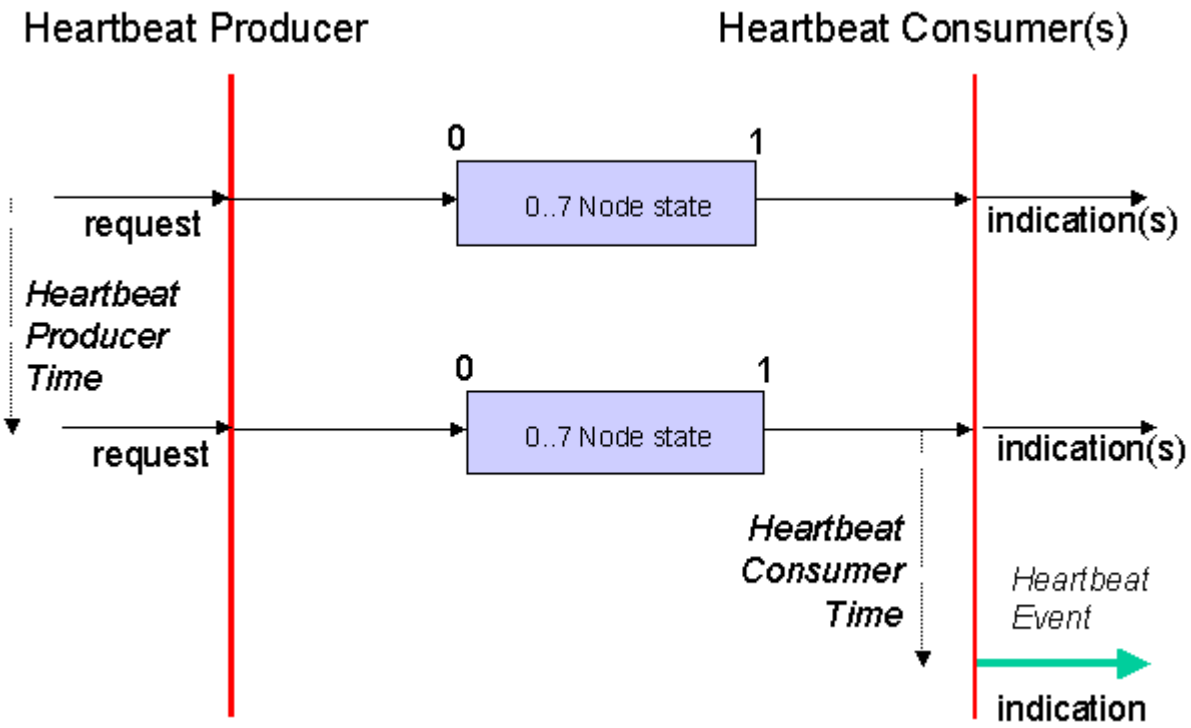


Abb. 35: Schematische Darstellung „Heartbeat-Verfahren“

Protokoll

Beim Heartbeat-Verfahren wird auf das Toggle-Bit verzichtet, die Knoten senden zyklisch Ihren Status (s). Siehe Guarding [▶ 40].

5.2 BootUp der FC510x

Einleitung

Die Firmware der FC510x CANopen PCI Karte behandelt jeden einzelnen Knoten individuell. Nach dem Systemstart wird zunächst geprüft, ob die erwarteten Knoten vorhanden sind und grundsätzlich den konfigurierten Geräten entsprechen. Danach wird jeder Knoten zunächst unabhängig von den anderen parametriert und gestartet. Im Folgenden wird das Aufstartverhalten für einen Knoten beschrieben.

1. Reset All Nodes

Die Aufstart-Sequenz beginnt mit einem globalen Reset Communication Telegramm, um alle Knoten in einen definierten Ausgangszustand zu bringen

2. Identify Node

Durch SDO Upload des Objekts 0x1000 (Device Type) wird zunächst festgestellt, ob der Knoten vorhanden ist. Dabei wird der vom Knoten gelieferte Inhalt auf Übereinstimmung mit dem erwarteten Wert überprüft. Objekt 0x1000 setzt sich aus Profilnummer und Additional Info zusammen, beide Werte sind im Karteireiter CAN Node zu finden.

Sind sowohl Additional Info als auch Profilnummer auf "0" gesetzt, so wird der zurückgelieferte Inhalt des Objektes 0x1000 nicht überprüft. Eine Antwort mit einem SDO Abort Protokoll wird nicht toleriert; dann wird abgebrochen.

Bei Werten ungleich null erfolgt der nächste Schritt nur bei Übereinstimmung. Ansonsten wird mit Node State 0x04 (SDO Syntax Error at StartUp, bei SDO Abort Protokoll oder falscher Datenlänge) bzw. 0x05 (SDO data mismatch at StartUp, bei fehlender Übereinstimmung) und entsprechender Fehlermeldung im Diag Fenster abgebrochen.

Wenn der Knoten nicht auf das SDO Upload Telegramm antwortet, so wird dieses SDO Protokoll nach Timeout (ca. 2 sec) abgebrochen und dann nach einer Wartezeit (ca. 1 sec) wiederholt, bis der Knoten antwortet. In dieser Phase steht der Node State auf 0x02 (Node not found).

Wenn im Karteireiter CAN Node die Vendor ID, der Product-Code, die Serial No. oder die Revision No. ungleich null konfiguriert wurden, so werden nun die entsprechenden Werte im Objekt 0x1018 des Knotens ausgelesen und verglichen. Nur bei Übereinstimmung wird der Bootup fortgesetzt.

3. Set SYNC Time

Falls synchrone PDOs konfiguriert wurden wird nun versucht, die vorgesehene Sync-Cycle-Time in Objekt 0x1006 (SYNC Intervall) einzutragen. Da dieses Objekt optional ist, wird der Boot-Up auch bei negativer Quittierung durch den Knoten fortgesetzt - eine Antwort des Knotens ist allerdings erforderlich.

4. Set PDO Parameter

Falls die Check-Box "Auto-Download der PDO Parameter" im Karteireiter CAN Node angewählt ist (Default), so werden nun die PDO Parameter aller konfigurierten PDOs geschrieben. Es sind dies der Identifier und der Transmission Type. Inhibit Time und Event Time werden dabei nur geschrieben, wenn sie ungleich null konfiguriert wurden.

Wenn der Knoten auf einen SDO Download der PDO Parameter mit einem SDO Abort Protokoll antwortet, so wird der entsprechende Eintrag anschließend gelesen (SDO Upload) und mit dem zu schreibenden Wert verglichen. Bei Übereinstimmung wird fortgesetzt. Auf diese Art können auch read-only PDO Parameter toleriert werden, wenn sie mit den konfigurierten Werten übereinstimmen.

Nur falls der Download bzw. der Vergleich mit den vorhandenen Werten erfolgreich waren erfolgt der nächste Schritt. Ansonsten wird mit Node State 0x04 bzw. 0x05 und entsprechender Fehlermeldung im Diag Fenster abgebrochen.

5. Set Guarding/Heartbeat

Falls für die Guard Time ein Wert ungleich null konfiguriert ist, werden nun die entsprechenden Parameter im Knoten beschrieben. Da Heartbeat weniger Buslast als Guarding erzeugt wird zunächst versucht, diese Form der Knotenüberwachung auf dem CANopen-Knoten zu starten.

Heartbeat: Eingetragen werden die Guard Time als Producer Heartbeat Time (0x1017) und das Produkt aus (Guard Time x Life Time Factor) als Consumer Heartbeat Time (0x1016). Die FC510x Karte sendet dann zyklisch ihr Heartbeat Telegramm mit der kleinsten konfigurierten Guard Time (die Guard Times können ja für jeden Knoten individuell eingestellt werden). Falls der Knoten den Eintrag der Consumer Heartbeat Time verweigert, so wird angenommen, dass der Knoten die Überwachung des Masters nicht unterstützt - dieses wird toleriert. Falls auch der Eintrag der Producer Heartbeat Time fehlschlägt, so wird das Guarding Protokoll konfiguriert.

Guarding: Falls der Knoten kein Heartbeat unterstützt werden die Guarding Parameter (Guard Time, 0x100C und Life Time Factor, 0x100D) eingetragen.

Schlägt auch dieser Versuch fehl, so wird der Aufstartvorgang mit Node State 0x04 und entsprechender Fehlermeldung im Diag Fenster abgebrochen.

6. Download User Parameter

Nun werden die im Karteireiter SDOs manuell hinzugefügten Objekte per SDO Download zum Knoten übertragen. Auch hier wird der Wert bei SDO Abbruch zurückgelesen und auf Übereinstimmung geprüft, um read-only Parameter zu tolerieren. Nur bei Erfolg wird fortgesetzt, ansonsten abgebrochen.

7. Start Node

Nach erfolgreichem Download aller Parameter wird der Knoten durch ein individuelles Start_Remote_Node Telegramm in den Zustand Operational überführt. Ca. 1 s nach diesem Start-Telegramm werden die RxPDOs erstmals an den Knoten geschickt und das Guarding bzw. Heartbeat Protokoll begonnen. Die Knotenüberwachung durch Heartbeat wird dabei erst begonnen, nachdem das Producer Heartbeat Telegramm des Knotens erstmals empfangen wurde.

Da CANopen keine explizite Bestätigung des Startvorgangs vorsieht, kann nur das erstmalige Eintreffen der Transmit PDOs ausgewertet werden. Bis alle konfigurierten TxPDOs eingetroffen sind, bleibt daher der Node State des Knotens auf 0x17 (Expected TxPDO is missing).

Nachdem alle konfigurierten Knoten gefunden, erfolgreich parametriert und individuell gestartet wurden, sendet die FC510x Karte noch ein globales Start_Remote_Node Telegramm (mit Node-ID=0).

8. SYNC

SYNC Telegramme werden erst geschickt, wenn die höchstpriore verknüpfte Task gestartet wurde. Synchrone TxPDOs werden daher auch erst getriggert, sobald diese Task läuft - auch dies kann eine Ursache für den Node State 0x17 sein.

Beispiel für eine Boot-Up Sequenz:

Knoten mit Node-ID1, Identifier im hex-Code.

```
Zeit   ID DLC DATA                    Desrciption
0.1244 00 2  82 00                    Reset communication all nodes
    Alle Knoten werden in Ausgangszustand versetzt
0.1252 601 8 40 00 10 00 00 00 00 00  [1000,00] Initiate Upload Rq.
   Erster Versuch, den Knoten 1 zu finden - Knoten ist noch im Reset
2.1316 601 8 80 00 00 00 00 00 04 05 05040000 [0000,00] Abort: SDO protocol timed out
   Knoten hat innerhalb SDO Time-Out (2 sec) nicht geantwortet, SDO wird abgebrochen
2.7875 701 1 00                       Boot-up
   Knoten hat Reset durchgeführt und meldet sich mit Boot-Up Nachricht
4.1391 601 8 40 00 10 00 00 00 00 00 [1000,00] Initiate Upload Rq.
   Zweiter Versuch, den Knoten 1 zu finden. Lesezugriff auf Objekt 0x1000
4.1411 581 8 43 00 10 00 91 01 07 00 91 01 07 00 [1000,00] Initiate Upload Rsp. expedited
   Knoten 1 antwortet mit Profile No. 0x191 (401dez) und Add. Info 0x07
4.1418 601 8 40 18 10 01 00 00 00 00 [1018,01] Initiate Upload Rq.
   Die Vendor ID wird ausgelesen
4.1434 581 8 43 18 10 01 02 00 00 00 02 00 00 00 [1018,01] Initiate Upload Rsp. expedited
   Knoten 1 antwortet mit Vendor ID 0x02 (= Beckhoff)
4.1442 601 8 23 00 18 01 81 01 00 00 81 01 00 00 [1800,01] Initiate Download Rq. expedited
   Nun wird der Identifier für TxPDO1 geschrieben: 0x181
4.1831 581 8 60 00 18 01 00 00 00 00 00 [1800,01] Initiate Download Rsp
   Knoten 1 bestätigt den Download
4.1840 601 8 23 01 18 01 81 02 00 00 81 02 00 00 [1801,01] Initiate Download Rq. expedited
   Identifier für TxPDO2 ist 0x281
4.2223 581 8 60 01 18 01 00 00 00 00 [1801,01] Initiate Download Rsp
4.2230 601 8 23 00 14 01 01 02 00 00 01 02 00 00 [1400,01] Initiate Download Rq. expedited
    Identifier für RxPDO1 ist 0x201
4.2347 581 8 60 00 14 01 00 00 00 00 [1400,01] Initiate Download Rsp
4.2356 601 8 2f 00 18 02 ff 00 00 00 ff [1800,02] Initiate Download Rq. expedited
   Transmission Type für TxPRO1 ist 0xFF=255
4.2737 581 8 60 00 18 02 00 00 00 00 [1800,02] Initiate Download Rsp
4.2744 601 8 2f 01 18 02 ff 00 00 00 ff [1801,02] Initiate Download Rq. expedited
   Transmission Type für TxPRO2 ist 0xFF=255
4.3133 581 8 60 01 18 02 00 00 00 00 [1801,02] Initiate Download Rsp
4.3141 601 8 2f 00 14 02 ff 00 00 00 ff [1400,02] Initiate Download Rq. expedited
   Transmission Type für RxPRO1 ist 0xFF=255
4.3252 581 8 60 00 14 02 00 00 00 00 [1400,02] Initiate Download Rsp
4.3264 601 8 2b 17 10 00 64 00 00 00 64 00 [1017,00] Initiate Download Rq. expedited
   Heartbeat Producer Time ist 0x64=100ms
4.3279 581 8 60 17 10 00 00 00 00 00 [1017,00] Initiate Download Rsp
4.3287 601 8 23 16 10 01 2c 01 7f 00 2c 01 7f 00 [1016,01] Initiate Download Rq. expedited
   Heartbeat Consumer Time ist 0x012C=300ms, Node ID des Heartbeat Producers (hier: FC5101) ist 0x7F
4.3304 581 8 60 16 10 01 00 00 00 00 [1016,01] Initiate Download Rsp
4.3312 601 8 23 00 55 00 00 00 ff ff 00 00 ff ff [5500,00] Initiate Download Rq. expedited
   User Parameter: Index 0x5500, SI 0, Wert 0x00 00 FF FF
4.3321 701 1 7f                       T0 Preoperational
   Knoten 1 setzt erstes Heartbeat Telegramm ab, FC5101 beginnt die Überwachung
4.4679 581 8 60 00 55 00 00 00 00 00 [5500,00] Initiate Download Rsp
4.4686 601 8 2f 23 64 00 01 00 00 00 01 [6423,00] Initiate Download Rq. expedited
   User Parameter: Index 0x6423, SI 0, Wert 0x01
4.4700 581 8 60 23 64 00 00 00 00 00 [6423,00] Initiate Download Rsp
4.4707 00 2 01 01                     Start Node
   Knoten 1 wird individuell in Operational überführt
4.4717 701 1 7f                       T0 Preoperational
   Das nächste Heartbeat Telegramm wird abgesetzt bevor der Statusübergang abgeschlossen ist
4.4986 181 1 00 00
   Knoten 1 ist Operational und sendet sein TxPDO1 und TxPDO2
4.4989 281 4 00 00 00 00 00 00 00 00
4.5786 701 1 05                       T0 Operational
4.6390 281 4 00 00 08 00 00 00 08 00
4.6411 281 4 00 00 00 00 00 00 00 00
4.6891 701 1 05                       T0 Operational
4.7951 701 1 05                       T0 Operational
4.9032 701 1 05                       T0 Operational
5.0048 281 4 00 00 08 00 00 00 08 00
5.0070 281 4 00 00 00 00 00 00 00 00
5.0094 701 1 05                       T0 Operational
5.0153 281 4 00 00 08 00 00 00 08 00
5.0174 281 4 00 00 00 00 00 00 00 00
5.1129 701 1 05                       T0 Operational
....
5.4755 00 2 01 00                     Start all nodes
   Nun werden alle Knoten gestartet
5.4847 201 1 00 00
   ca. 1 sec nach Start von Knoten 1 wird das RxPDO1 erstmals geschickt
```

5.3 Prozessdatenobjekte (PDO)

Einführung

Bei vielen Feldbus-Systemen wird ständig das gesamte Prozessabbild übertragen - meist mehr oder weniger zyklisch. CANopen ist nicht auf dieses Kommunikationsprinzip beschränkt, da CAN durch die Multi-Master Buszugriffsregelung auch andere Möglichkeiten bietet: die Prozessdaten werden bei CANopen nicht im Master/Slave-Verfahren übertragen, sondern folgen dem Produzenten/Konsumenten-Modell (Producer/ Consumer). Hierbei sendet ein Busknoten seine Daten von sich aus (Producer), beispielsweise durch den Eintritt eines Ereignisses getriggert; alle anderen Knoten hören mit und entscheiden anhand des Identifiers, ob sie sich für dieses Telegramm interessieren und verarbeiten es entsprechend (Consumer).

Bei CANopen werden die Prozessdaten in Segmente zu maximal 8 Byte aufgeteilt. Diese Segmente heißen Prozessdatenobjekte (PDOs). Die PDOs entsprechen jeweils einem CAN-Telegramm und werden über dessen spezifischen CAN-Identifier zugeordnet und in ihrer Priorität bestimmt. Man unterscheidet Empfangs-PDOs (Receive-PDOs , RxPDOs) und Sende-PDOs (Transmit-PDOs , TxPDOs), wobei die Bezeichnung jeweils aus Gerätesicht erfolgt: eine Ein-/Ausgabebaugruppe sendet ihre Eingangsdaten mit TxPDOs, und empfängt die Ausgangsdaten in den RxPDOs. **Diese Bezeichnung wird im TwinCAT-System-Manager beibehalten.**

Kommunikationsparameter

Die PDOs können je nach Applikationsanforderung mit unterschiedlichen Kommunikationsparametern versehen werden. Wie alle CANopen-Parameter stehen auch diese im Objektverzeichnis des Gerätes, auf sie kann über die Servicedatenobjekte zugegriffen werden. Die Parameter für die Empfangs-PDOs stehen bei Index 0x1400 (RxPDO1) und folgende, bis zu 512 RxPDOs können vorhanden sein (Bereich bis Index 0x15FF). Entsprechend finden sich die Einträge für die Sende-PDOs bei Index 0x1800 (TxPDO1) bis 0x19FF (TxPDO512).

Für den Prozessdatenaustausch stehen auf den Beckhoff Buskopplern bzw. Feldbus Koppler Box Baugruppen jeweils 16 RxPDO und TxPDOs zur Verfügung (bei den Economy- und LowCost-Kopplern BK5110 und LC5100 sowie den Feldbus Boxen sind es jeweils 5 PDOs, da diese Geräte über weniger Prozessdaten verfügen). Die FC510x CANopen Master Karte unterstützt - beschränkt durch die DPRAM-Größe - je Kanal bis zu 192 Sende- und 192 Empfangs-PDOs. Die CANopen Klemme EL6751 organisiert das Prozessabbild dynamisch, d.h. die Prozessdaten werden hintereinander geschrieben, was eine höhere Datenübertragungsrate ermöglicht. Im Slave Mode können bis zu 32 TxPDOs und 32 RxPDOs verarbeitet werden.

Für jedes vorhandene Prozessdatenobjekt ist ein zugehöriges Kommunikationsparameter-Objekt vorhanden. Der TwinCAT-System-Manager ordnet die eingestellten Parameter automatisch den jeweiligen Objektverzeichniseinträgen zu. Im Folgenden werden diese Einträge samt ihrer Bedeutung für das Kommunikationsverhalten der Prozessdaten erläutert.

PDO-Identifier

Der wichtigste Kommunikationsparameter eines PDOs ist der CAN-Identifier (auch Communication Object Identifier, COB-ID genannt). Er dient zur Identifizierung der Daten und bestimmt deren Priorität beim Buszugriff. Für jedes CAN-Datentelegramm darf es nur einen Sendeknoten (Producer) geben; da CAN jedoch alle Nachrichten im Broadcast-Verfahren sendet kann ein Telegramm wie beschrieben von beliebig vielen Knoten empfangen werden (Consumer). Ein Knoten kann also seine Eingangsinformation mehreren Busteilnehmern gleichzeitig zur Verfügung stellen - auch ohne Weiterleitung durch einen logischen Busmaster. Der Identifier steht in Subindex 1 des Kommunikationsparametersatzes. Er ist als 32-Bit Wert kodiert, wobei die niederwertigsten 11 Bits (Bit 0...10) den eigentlichen Identifier enthalten. Die Datenbreite des Objektes von 32 Bit erlaubt auch den Eintrag von 29 Bit Identifiern nach CAN 2.0B, allerdings beziehen sich die Default-Identifier stets auf die üblichere 11 Bit-Variante. Allgemein geht CANopen sparsam mit den zur Verfügung stehenden Identifiern um, sodass der Einsatz der 29 Bit-Variante auf Sonderanwendungen beschränkt bleibt - und daher auch von den Beckhoff CANopen Geräten nicht unterstützt wird. Über das höchstwertige Bit (Bit 31) lässt sich das Prozessdatenobjekt aktivieren bzw. abschalten.

Im Anhang finden Sie eine komplette Identifier-Liste [▶ 81].

PDO Linking

Im System der Default-Identifier kommunizieren alle Knoten (hier: Slaves) mit einer Zentrale (Master), da kein Slave-Knoten per Default auf die Sende-Identifier eines anderen Slave-Knotens hört).

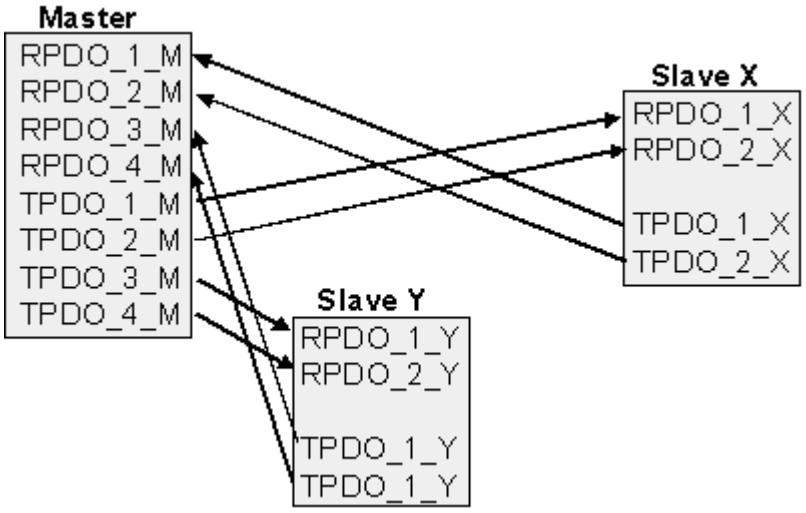


Abb. 36: Default Identifier-Verteilung: Master/Slave

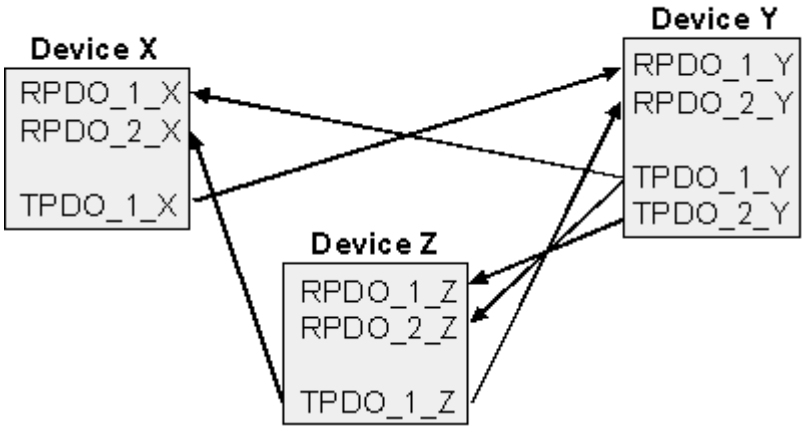


Abb. 37: PDO Linking: Peer to Peer

Wenn das Consumer-Producer-Modell der CANopen PDOs zum direkten Datenaustausch zwischen Knoten (ohne Master) genutzt werden soll, so muss die Identifier-Verteilung entsprechend angepasst werden, damit der TxPDO-Identifier des Producers mit dem RxPDO-Identifier des Consumers übereinstimmt. Dieses Verfahren nennt man PDO Linking. Es ermöglicht beispielsweise den einfachen Aufbau von elektronischen Getrieben, bei denen mehrere Slave-Achsen gleichzeitig auf den Ist-Wert im TxPDO der Master-Achse hören.

PDO-Kommunikationsarten: Überblick

CANopen bietet vielfältige Möglichkeiten, die Prozessdaten zu übertragen (siehe auch: Hinweise zur PDO Parametrierung [▶ 51])

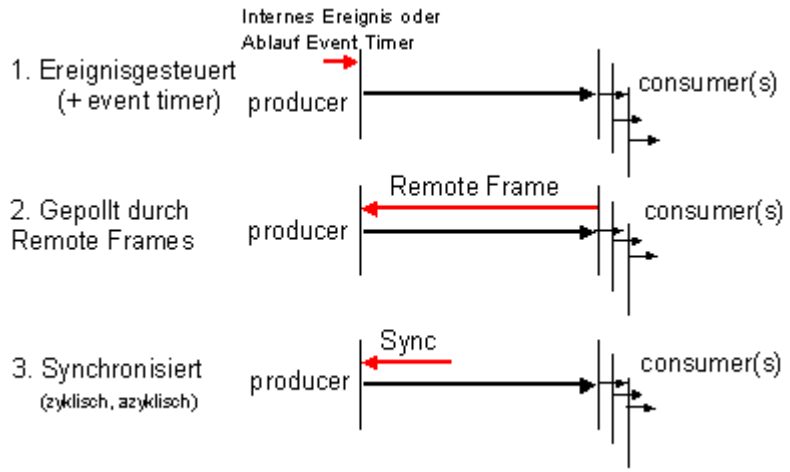


Abb. 38: Darstellung Übertragung CAN-Prozessdaten

Ereignisgesteuert

Das "Ereignis" ist die Änderung eines Eingangswertes, die Daten werden sofort nach dieser Änderung verschickt. Durch die Ereignissteuerung wird die Busbandbreite optimal ausgenutzt, da nicht ständig das Prozessabbild, sondern nur die Änderung desselben übertragen wird. Gleichzeitig wird eine kurze Reaktionszeit erreicht, da bei Änderung eines Eingangswertes nicht erst auf die nächste Abfrage durch einen Master gewartet werden muss.

Ab CANopen Version 4 kann die ereignisgesteuerte Kommunikationsart mit einem zyklischen Update kombiniert werden. Auch wenn gerade kein Ereignis aufgetreten ist, werden ereignisgesteuerte TxPDO nach Ablauf des Event Timers verschickt. Beim Auftreten eines Ereignisses wird der Event Timer zurückgesetzt. Bei RxPDOs wird der Event Timer als Watchdog benutzt um das Eintreffen von ereignisgesteuerten PDOs zu überwachen. Sollte innerhalb der eingestellten Zeit kein PDO eingetroffen sein, so geht der Busknoten in den Fehlerzustand.

Gepollt

Die PDOs können auch durch Datenanforderungstelegramme (Remote Frames) gepollt werden. Auf diese Art kann etwa das Eingangsprozessabbild bei ereignisgesteuerten Eingängen auch ohne deren Änderung auf den Bus gebracht werden, beispielsweise bei einem zur Laufzeit ins Netz aufgenommenen Monitor- oder Diagnosegerät. Das zeitliche Verhalten von Remote Frame und Antworttelegramm hängt von den verwendeten CAN-Controllern ab. Bausteine mit integrierter kompletter Nachrichtenfilterung ("FullCAN") beantworten ein Datenanforderungstelegramm in der Regel direkt und versenden sofort die im entsprechenden Sendebuffer stehenden Daten - dort muss die Applikation dafür Sorge tragen, dass die Daten ständig aktualisiert werden. CAN-Controller mit einfacher Nachrichtenfilterung (BasicCAN) reichen die Anforderung dagegen an die Applikation weiter, die nun das Telegramm mit den aktuellen Daten zusammenstellen kann. Das dauert länger, dafür sind die Daten aktuell. Beckhoff verwendet CAN Controller nach dem Basic CAN Prinzip.

Da dieses Geräteverhalten für den Anwender meist nicht transparent ist und zudem noch CAN-Controller in Verwendung sind, die Remote Frames überhaupt nicht unterstützen, kann die gepollte Kommunikationsart nur bedingt für den laufenden Betrieb empfohlen werden.

Synchronisiert

Nicht nur bei Antriebsanwendungen ist es sinnvoll, das Ermitteln der Eingangsinformation sowie das Setzen der Ausgänge zu synchronisieren. CANopen stellt hierzu das SYNC-Objekt zur Verfügung, ein CAN-Telegramm hoher Priorität ohne Nutzdaten, dessen Empfang von den synchronisierten Knoten als Trigger für das Lesen der Eingänge bzw. für das Setzen der Ausgänge verwendet wird.

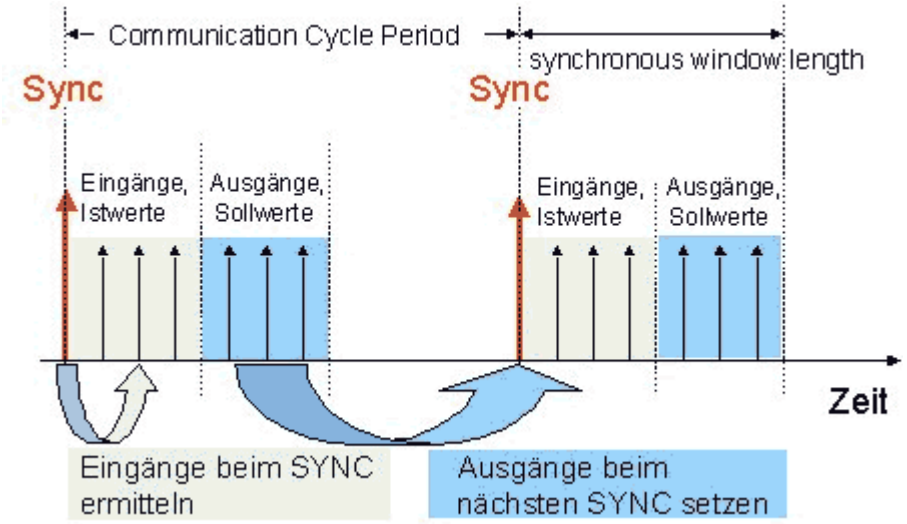


Abb. 39: Darstellung CAN Telegramm „SYNC“

PDO-Übertragungsart: Parametrierung

Der Parameter PDO-Übertragungsart (Transmission Type) legt fest, wie das Versenden des PDOs ausgelöst wird bzw. wie empfangene PDOs behandelt werden:

Übertragungsart	Zyklisch	Azyklisch	Synchron	Asynchron	Nur RTR
0		X	X		
1-240	X		X		
241-251	- reserviert -				
252			X		X
253				X	X
254, 255				X	

Die Übertragungsart wird für RxPDOs in den Objekten 0x1400ff, Subindex 2, und für TxPDOs in den Objekten 0x1800ff, Subindex 2 parametriert.

Azyklisch Synchron

PDOs der Übertragungsart 0 arbeiten synchron, aber nicht zyklisch. Ein RxPDO wird erst nach Empfang des nächsten SYNC-Telegramms ausgewertet. Damit lassen sich beispielsweise Achsgruppen nacheinander mit neuen Zielpositionen versehen, die alle beim nächsten SYNC gültig werden - ohne dass ständig Stützstellen ausgegeben werden müssen. Ein Gerät, dessen TxPDO auf Übertragungsart 0 konfiguriert ist, ermittelt seine Eingangsdaten beim Empfang des SYNC (synchrones Prozessabbild) und sendet sie anschließend, falls die Daten einem Ereignis entsprechen (beispielsweise eine Eingangsänderung) eingetreten ist. Die Übertragungsart 0 kombiniert also den Sendegrund "ereignisgesteuert" mit dem Sende- (und möglichst Sample-) bzw. Verarbeitungs-Zeitpunkt "SYNC-Empfang".

Zyklisch Synchron

Bei Übertragungsart 1-240 wird das PDO zyklisch gesendet: nach jedem "n-ten" SYNC (n=1...240). Da die Übertragungsart nicht nur im Netz, sondern auch auf einem Gerät kombiniert werden dürfen, kann so z. B. ein schneller Zyklus für digitale Eingänge vereinbart werden (n=1), während die Daten der Analogeingänge in einem langsameren Zyklus übertragen werden (z. B. n=10). RxPDOs unterscheiden in der Regel nicht zwischen den Übertragungsarten 0...240: ein empfangenes PDO wird beim nächsten SYNC-Empfang gültig gesetzt. Die Zykluszeit (SYNC-Rate) kann überwacht werden (Objekt 0x1006), das Gerät reagiert bei SYNC-Ausfall dann entsprechend der Definition des Geräteprofils und schaltet z. B. seine Ausgänge in den Fehlerzustand.

Die FC510x Karte / EL6751Klemme unterstützen die synchrone Kommunikationsart vollständig: das Versenden des SYNC Telegramms ist mit der verknüpften Task gekoppelt, sodass zu jedem Taskbeginn neue Eingangsdaten zur Verfügung stehen. Das Ausbleiben eines synchronen PDOs wird erkannt und an die Applikation gemeldet.

Nur RTR

Die Übertragungsarten 252 und 253 gelten für Prozessdatenobjekte, die ausschließlich auf Anforderung durch ein Remote Frame übertragen werden. 252 ist synchron: beim Empfang des SYNCs werden die Prozessdaten ermittelt, gesendet werden sie nur auf Anforderung. 253 ist asynchron, hier werden die Daten ständig ermittelt und auf Anforderung verschickt. Diese Übertragungsart ist generell nicht zu empfehlen, da das Abholen der Eingangsdaten von einigen CAN Controllern nur unvollständig unterstützt wird. Da die CAN Controller zudem teilweise selbsttätig auf Remote Frames antworten (ohne vorher aktuelle Eingangs-Daten anzufordern), ist die Aktualität der gepollten Daten unter Umständen fragwürdig. Die Übertragungsart 252 und 253 wird aus diesen Gründen von den Beckhoff PC-Karten / Klemmen nicht unterstützt.

Asynchron

Die Übertragungsarten 254 + 255 sind asynchron oder auch ereignisgesteuert. Bei Übertragungsart 254 ist das Ereignis herstellerspezifisch, bei 255 im Geräteprofil definiert. Im einfachsten Fall ist das Ereignis die Veränderung eines Eingangswertes - es wird also jede Werteänderung übertragen. Die Asynchrone Übertragungsart kann mit dem Event Timer gekoppelt werden und liefert so auch dann Eingangsdaten, wenn aktuell kein Ereignis aufgetreten ist.

Inhibit Zeit

Über den Parameter "Inhibit-Zeit" kann ein "Sende-Filter" aktiviert werden, der die Reaktionszeit bei der relativ ersten Eingangsänderung nicht verlängert, aber bei unmittelbar darauffolgenden Änderungen aktiv ist. Die Inhibit-Zeit (Sendeverzögerungszeit) beschreibt die Zeitspanne, die zwischen dem Versenden zweier gleicher Telegramme mindestens abgewartet werden muss. Wenn die Inhibit-Zeit genutzt wird, so kann die maximale Busbelastung und damit die Latenzzeit im "worst case"-Fall ermittelt werden.

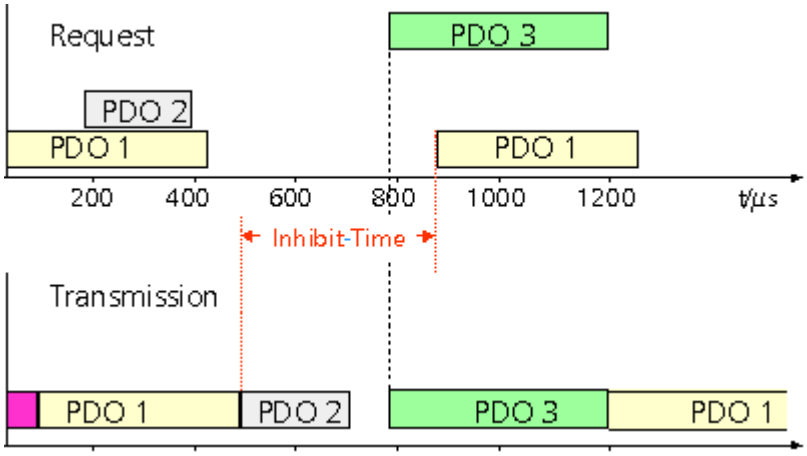


Abb. 40: Zeitl. Diagramm „Inhibit-Time“

Die Beckhoff PC-Karten FC510x / EL6751 Klemme können zwar die Inhibit-Zeit auf Slave-Geräten parametrieren, unterstützen sie jedoch selbst nicht. Eine Spreizung der gesendeten PDOs (Sendeverzögerung) ergibt sich automatisch aus der gewählten Zyklus-Zeit der SPS - und es macht wenig Sinn, die SPS schneller laufen zu lassen als es die Busbandbreite zulässt. Zudem kann die Busbelastung wirkungsvoll über die synchrone Kommunikation beeinflusst werden.

Event Timer

Über Subindex 5 der Kommunikationsparameter lässt sich ein Ereignis-Timer (Event Timer) für Sende-PDOs festlegen. Der Ablauf dieses Timers wird als zusätzlich eingetretenes Ereignis für das entsprechende PDO gewertet, das PDO wird also dann gesendet. Wenn das Applikationsereignis während einer Timer-Periode auftritt, so wird ebenfalls gesendet und der Timer wird zurückgesetzt.

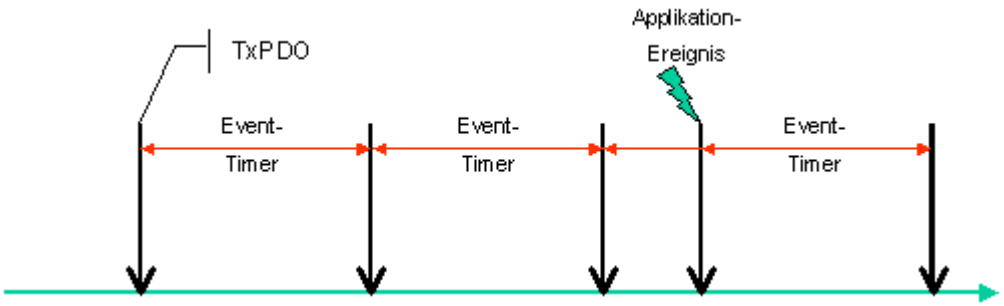


Abb. 41: Zeitliche Darstellung des Event-Timers

Bei Empfangs-PDOs wird der Timer-Parameter dazu verwendet, die Überwachungszeit für dieses PDO anzugeben: Die Applikation wird benachrichtigt, wenn kein entsprechendes PDO innerhalb der eingestellten Zeit empfangen wurde. Auf diese Art kann die FC510x / EL6751 jedes einzelne PDO individuell überwachen.

Hinweise zur PDO Parametrierung [▶ 51]

PDO Mapping

Unter PDO-Mapping versteht man die Abbildung der Applikationsobjekte (Echtzeitdaten) aus dem Objektverzeichnis in die Prozessdatenobjekte. Die CANopen-Geräteprofile sehen für jeden Gerätetyp ein Default Mapping vor, das für die meisten Anwendungen passend ist. So bildet das Default Mapping für digitale E/A einfach die Ein- bzw. Ausgänge ihrer physikalischen Reihenfolge gemäß in die Sende- bzw. Empfangs-Prozessdatenobjekte ab.

Die Default-PDOs für Antriebe enthalten jeweils 2 Byte Steuer- bzw. Statuswort und Soll- bzw. Istwert für die betreffende Achse.

Das aktuelle Mapping kann über entsprechende Einträge im Objektverzeichnis, die sogenannten Mapping-Tabellen, gelesen werden. An erster Stelle der Mapping Tabelle (Subindex 0) steht die Anzahl der gemappten Objekte, die im Anschluss aufgelistet sind. Die Tabellen befinden sich im Objektverzeichnis bei Index 0x1600 ff. für die RxPDOs bzw. 0x1A00ff für die TxPDOs.

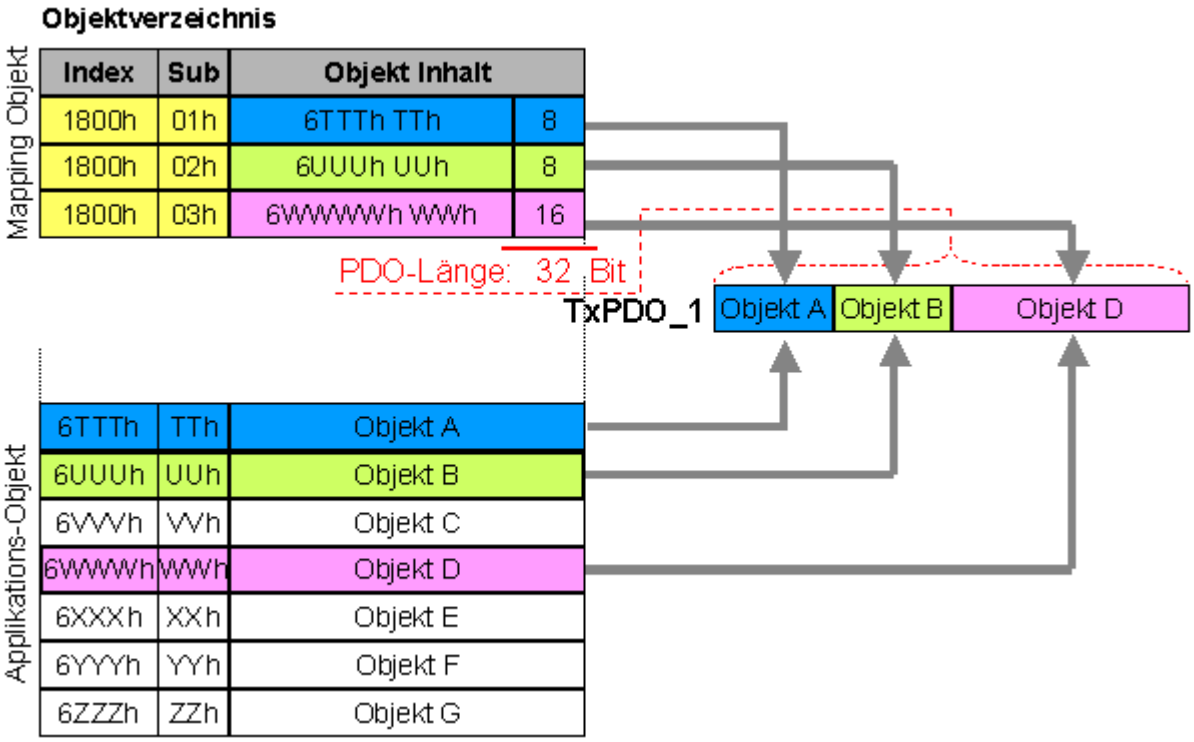


Abb. 42: Darstellung Mapping

Digitale und analoge Ein-/Ausgabebaugruppen: E/A-Anzahl auslesen

Die aktuelle Anzahl der digitalen und analogen Ein-/Ausgänge lässt sich durch Auslesen der entsprechenden Applikationsobjekte im Objektverzeichnis ermitteln bzw. verifizieren:

Parameter	Adresse Objektverzeichnis
Anzahl digitale Eingangsbytes	Index 0x6000, Subindex 0
Anzahl digitale Ausgangsbytes	Index 0x6200, Subindex 0
Anzahl analoge Eingänge	Index 0x6401, Subindex 0
Anzahl analoge Ausgänge	Index 0x6411, Subindex 0

Variables Mapping

In der Regel genügt die Default-Belegung der Prozessdatenobjekte (Default Mapping) bereits den Anforderungen. Für spezielle Anwendungsfälle kann die Belegung jedoch verändert werden: So unterstützen beispielsweise die Beckhoff CANopen Buskoppler das variable Mapping, bei dem die Applikationsobjekte (Ein- und Ausgangsdaten) frei den PDOs zugeordnet werden können. Hierzu müssen die Mapping-Tabellen konfiguriert werden: Ab CANopen Version 4 ist nur noch die folgende Vorgehensweise zulässig, die genau eingehalten werden muss:

1. Zunächst PDO löschen (0x1400ff, bzw. 0x1800ff, Subindex 1, Bit 31 auf "1" setzen)
2. Subindex 0 im Mapping Parameter (0x1600ff bzw. 0x1A00ff) auf "0" setzen
3. Mapping Einträge (0x1600ff bzw. 0x1A00ff, SI 1..8) verändern
4. Subindex 0 im Mapping Parameter auf gültigen Wert setzen. Das Gerät überprüft dann die Einträge auf Konsistenz.
5. PDO anlegen durch Eintragen d. Identifiers (0x1400ff bzw. 0x1800ff Subindex 1).

Dummy-Mapping

Ein weiteres Feature von CANopen ist das Mappen von Platzhaltern (Dummy-Einträgen). Als Platzhalter dienen die im Objektverzeichnis hinterlegten Datentyp-Einträge, die ja selbst nicht mit Daten versehen sind. Sind solche Einträge in der Mapping-Tabelle enthalten, so werden die entsprechenden Daten vom Gerät nicht ausgewertet. Auf diese Art können beispielsweise mehrere Antriebe über ein einziges CAN-Telegramm mit neuen Sollwerten versorgt werden oder Ausgänge auf mehreren Knoten auch im ereignisgesteuerten Modus gleichzeitig gesetzt werden.

5.4 PDO-Parametrierung

Auch wenn die meisten CANopen-Netze in der Default-Einstellung und damit mit minimalem Konfigurationsaufwand zufrieden stellend arbeiten, so sollte zumindest überprüft werden, ob die vorhandene Buslast vertretbar ist. 80% Busauslastung mag für ein rein zyklisch synchron arbeitendes Netzwerk akzeptabel sein, für ein rein ereignisgesteuertes Netz ist dieser Wert in der Regel zu hoch, da kaum Bandbreite für zusätzliche Ereignisse zur Verfügung steht.

Applikationsanforderungen berücksichtigen

Die Prozessdatenkommunikation sollte hinsichtlich einiger sich teilweise widersprechender Applikationsanforderungen optimiert werden. Hierzu gehören

- Geringer Parametrieraufwand - optimal sind brauchbare Default-Werte
- Garantierte Reaktionszeit auf bestimmte Ereignisse
- Zykluszeit bei Regelvorgängen über den Bus
- Sicherheitsreserven für Busstörungen (genügend Bandbreite für Nachrichtenwiederholung)
- Maximale Baud-Rate - hängt von der maximalen Buslänge ab
- Gewünschte Kommunikationspfade - wer spricht mit wem

Der bestimmende Faktor ist meist die zur Verfügung stehende Busbandbreite (Buslast).

Baud-Rate

Allgemein wird man beginnen, die Baud-Rate so groß zu wählen, wie es die Buslänge erlaubt. Hierbei sollte man berücksichtigen, dass serielle Bussysteme grundsätzlich um so empfindlicher auf Störeinflüsse reagieren, je höher die Baud-Rate ist. Es gilt also die Regel: so schnell wie nötig. 1000 kBit/s sind meist nicht erforderlich und uneingeschränkt nur bei Netzwerken innerhalb eines Schaltschrankes ohne galvanische Trennung der Busknoten empfehlenswert. Die Erfahrung zeigt auch, dass das Abschätzen der verlegten Buskabellänge häufig zu optimistisch erfolgt - die tatsächliche Kabellänge also größer ist.

Kommunikationsart bestimmen

Ist die Baud-Rate gewählt, so gilt es nun die PDO-Kommunikationsart(en) zu bestimmen. Diese haben unterschiedliche Vor- und Nachteile:

- Die zyklisch synchrone Kommunikation ergibt eine genau vorhersagbare Busbelastung und damit ein definiertes Zeitverhalten - man könnte auch sagen, der worst case ist Standard. Sie ist einfach zu konfigurieren: mit dem Parameter SYNC-Rate kann die Buslast global eingestellt werden. Die Prozessabbilder werden synchronisiert: Eingänge werden gleichzeitig gelesen, Ausgangsdaten gleichzeitig gültig gesetzt - die Qualität dieser Synchronisierung ist allerdings implementierungsabhängig. Die BECKHOFF PC-Karten FC510x / CANopen-Klemme EL6751sind in der Lage, das CANopen Bussystems mit den Zyklen der Anwendungsprogramme (SPS bzw. NC) zu synchronisieren.

  Die garantierte Reaktionszeit ist bei der zyklisch synchronen Kommunikation immer mindestens so groß wie die Zykluszeit, und die Busbandbreite wird nicht optimal genutzt, da auch alte, sich nicht ändernde Daten ständig übertragen werden. Es ist aber möglich, das Netz durch die Wahl unterschiedlicher SYNC-Vielfacher (Transmission Types 1...240) zu optimieren und sich langsam ändernde Daten seltener zu übertragen als z. B. zeitkritische Eingänge. Berücksichtigt werden sollte jedoch, dass Eingangszustände, die kürzer anstehen als die Zykluszeit, nicht unbedingt kommuniziert werden. Ist dies gefordert, so sollten die entsprechenden PDOs für asynchrone Kommunikation vorgesehen werden.
- Die ereignisgesteuerte, asynchrone Kommunikation ist optimal hinsichtlich Reaktionszeit und Verwendung der Busbandbreite - man könnte sie als "CAN pur" bezeichnen. Bei ihrer Wahl muss allerdings berücksichtigt werden, dass unter Umständen viele Ereignisse gleichzeitig auftreten und sich dann entsprechende Verzögerungszeiten einstellen können, bis ein relativ niederpriores PDO verschickt werden kann - eine seriöse Netzwerkplanung erfordert demnach eine worst-case Betrachtung. Auch muss, z. B. durch Verwendung der Inhibit Zeit [▶ 45], verhindert werden, dass ein sich ständig ändernder Eingang mit hoher PDO-Priorität den Bus blockiert (Fachbegriff: "babbling idiot"). Aus diesem Grund ist beispielsweise die Ereignissteuerung bei Analogeingängen im Geräteprofil per Default abgeschaltet und muss gezielt aktiviert werden. Über den Ablauf-Timer lassen sich Zeitfenster für die Sende-PDOs einstellen: Das Telegramm wird frühestens nach Ablauf der Inhibit-Zeit [▶ 45] und spätestens nach Verstreichen des Ablauf-Timers erneut gesendet.
- Parametriert wird die Kommunikationsart über den Transmission Type [▶ 45].

Es ist auch möglich, beide PDO Kommunikationsprinzipien zu kombinieren. So kann es beispielsweise sinnvoll sein, die Soll- und Istwerte einer Achsregelung zyklisch synchron auszutauschen, während Endschalter oder die mit Grenzwerten versehene Motortemperatur mit ereignisgesteuerten PDOs überwacht werden. So kombiniert man die Vorteile beider Prinzipien: Synchronität der Achskommunikation und kurze Reaktionszeit für Endschalter. Durch die dezentrale Grenzwertüberwachung wird trotz Ereignissteuerung vermieden, dass der Temperatur-Analogwert ständig zur Buslast beiträgt.

Im genannten Beispiel kann es auch sinnvoll sein, die Identifier-Verteilung gezielt zu beeinflussen, um den Buszugriff durch die Prioritätsverteilung zu optimieren: die höchste Priorität bekommt das PDO mit den Endschalterdaten, die niedrigste das mit den Temperaturwerten.

In aller Regel ist es aber nicht erforderlich, die Identifier-Verteilung anzupassen, um die Latenzzeit beim Buszugriff zu optimieren. Dagegen müssen die Identifier verändert werden, um eine masterlose Kommunikation zu ermöglichen (PDO Linking [▶ 45]). Im genannten Beispiel könnte je ein RxPDO der Achsen denselben Identifier wie das TxPDO des Endschalters zugewiesen bekommen und dadurch eine Veränderung des Eingangswertes verzögerungsfrei empfangen.

Buslast bestimmen

In jedem Fall ist es sinnvoll, die Buslast zu bestimmen. Doch welche Buslastwerte sind zulässig bzw. sinnvoll? Unterscheiden sollte man zunächst den kurzfristigen Burst von Telegrammen, bei dem eine Anzahl CAN-Nachrichten direkt aufeinander folgt - kurzzeitig 100% Buslast. Das ist nur dann problematisch, wenn die dadurch ausgelöste Folge von Empfangsinterrupts auf den CAN-Knoten nicht mehr abgearbeitet werden kann, es also zu einem Datenüberlauf (CAN-Queue-Overrun) kommt. Das kann bei sehr hohen Baud-Raten (> 500 kBit/s) bei Knoten mit Software-Telegrammfilterung und relativ langsamen oder stark ausgelasteten Mikro-Controllern vorkommen, wenn z. B. eine direkte Folge von Remote Frames (diese enthalten keine Datenbytes und haben daher minimale Länge) auf dem Bus ist (bei 1 Mbit/s kann so alle 40 µs ein Interrupt erzeugt werden; Beispiel: ein NMT-Master sendet alle Guarding-Anforderungen direkt hintereinander). Durch geschickte Implementierung läßt sich das vermeiden, der Anwender sollte davon ausgehen können, dass von den Geräteanbietern hierfür Sorge getragen wurde. Ein Burst-Zustand ist z. B. direkt nach dem SYNC Telegramm völlig normal: vom SYNC getriggert versuchen alle synchron arbeitenden Knoten quasi gleichzeitig Ihre Daten zu senden, es finden viele Arbitrierungsvorgänge statt, die Telegramme sortieren sich nacheinander in der Reihenfolge ihrer Priorität auf den Bus. Das ist in der Regel unkritisch, da es sich hier um Telegramme mit einigen Datenbytes handelt und die Telegrammfolge damit zwar eine schnelle, aber überschaubare Folge von Empfangsinterrupts auf den CAN-Knoten auslöst.

Unter Buslast versteht man meist den gemittelten Wert über mehrere Primärzyklen, also z. B. das Mittel über 100-500 ms. CAN, und damit CANopen, ist zwar in der Lage, nahe 100% Buslast auf Dauer zu bewältigen, aber dann steht keine Bandbreite für eventuelle Wiederholungen bei Störeinflüssen, asynchrone Fehlermeldungen, Parametrierung etc. zur Verfügung. Selbstverständlich hat die vorherrschende Art der Kommunikation einen großen Einfluss auf die sinnvolle Buslast: ein komplett zyklisch synchron arbeitendes Netz befindet sich ja bereits nahe am worst case Zustand und kann daher mit Werten von 70-80% betrieben werden. Für ein rein ereignisgesteuertes Netz ist diese Zahl nur schwer anzugeben: es muss hier abgeschätzt werden, wie viele zusätzliche Ereignisse im Vergleich zum derzeitigen Anlagenzustand auftreten können und für wie lange das zu einem Burst führt - also wie lange die relativ niederpriorste Nachricht dann verzögert würde. Ist dieser Wert von der Applikation her zulässig, so ist die aktuelle Buslast akzeptabel. Als Näherungswert kann meist angenommen werden, dass ein ereignisgesteuertes Netz mit 30-40% Grundlast genügend Reserven für worst-case-Szenarien hat - diese Annahme macht aber eine sorgfältige Analyse nicht überflüssig, wenn Verzögerungen zu kritischen Anlagenzuständen führen können.

Die BECKHOFF CANopen-Master-Karten FC510x / CANopen-Masterklemme EL6751 zeigen die Buslast über den System Manager ein. Diese Variable kann auch in der SPS verarbeitet oder in der Visualisierung zur Anzeige gebracht werden.

Neben den Kommunikationsparametern ist natürlich die Datenbelegung der Prozessdatenobjekte entscheidend: das PDO Mapping. [▶ 50]

5.5 Servicedatenobjekte (SDO)

Die im Objektverzeichnis aufgeführten Parameter werden über Servicedatenobjekte gelesen und beschrieben. Diese SDOs sind *Multiplexed Domains*, also Datenstrukturen beliebiger Größe, die mit einem Multiplexor (Adresse) versehen sind. Der Multiplexor besteht aus 16-Bit-Index und 8-Bit-Subindex, die die entsprechenden Einträge im Objektverzeichnis adressieren.

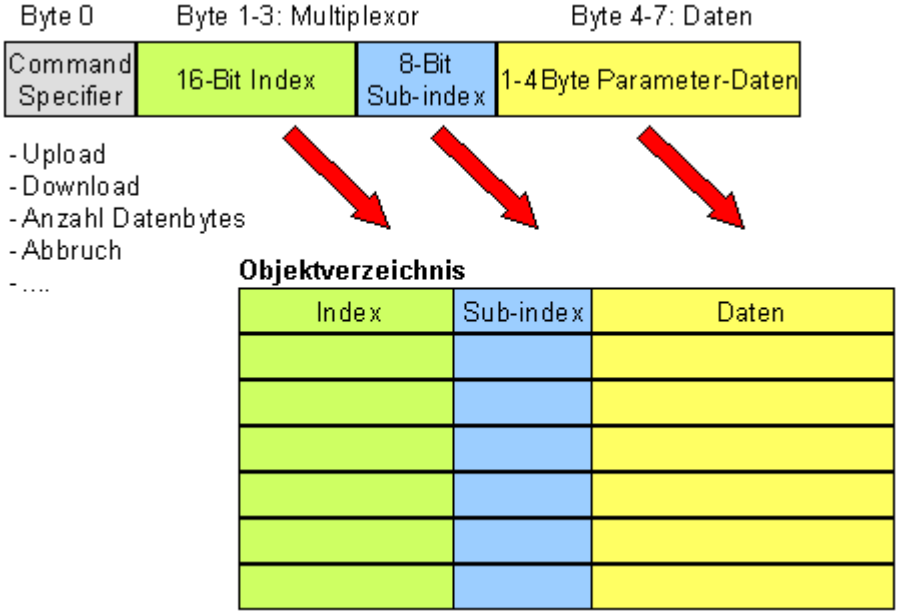


Abb. 43: SDO-Protokoll: Zugriff auf Objektverzeichnis

Die CANopen Buskoppler sind Server für das SDO, d.h. sie stellen auf Anforderung des Clients (z. B. des IPCs oder der SPS) Daten zur Verfügung (Upload) oder sie empfangen Daten vom Client (Download). Dabei findet ein Handshake zwischen Client und Server statt.

Wenn der zu übertragende Parameter bis zu 4 Bytes umfasst, genügt ein einziger Handshake (ein Telegrammpaar): Beim Download sendet der Client die Daten zusammen mit Index, Subindex und der Server bestätigt den Erhalt. Beim Upload fordert der Client die Daten an indem er Index und Subindex des gewünschten Parameters überträgt, und der Server sendet den Parameter (incl. Index und Subindex) in seinem Antworttelegramm.

Für Upload und Download wird das gleiche Identifier-Paar verwendet. In den stets 8 Byte großen Telegrammen sind im ersten Datenbyte die unterschiedlichen Dienste codiert. Bis auf die Objekte 1008h, 1009h und 100Ah (Gerätename, Hardware- bzw. Softwareversion) sind alle Parameter der Buskoppler nur bis zu 4 Byte groß, daher beschränkt sich diese Beschreibung auf die Übertragung dieser Daten im beschleunigten Transfer (Expedited Transfer).

Protokoll

Im Folgenden wird der Aufbau der SDO-Telegramme beschrieben.

Client -> Server, Upload Request

11-bit Identifier	8 Byte Nutzdaten							
0x600 (=1536dez) + Node-ID	0x40	Index0	Index1	SubIdx	0x00	0x00	0x00	0x00

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)

Client -> Server, Upload Response

11-bit Identifier	8 Byte Nutzdaten							
0x580 (=1408dez) + Node-ID	0x4x	Index0	Index1	SubIdx	Data0	Data1	Data2	Data3

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)
Data0	Daten Low-Low-Byte (LLSB)
Data3	Daten High-High-Byte (MMSB)

Parameter des Datentyps Unsigned8 werden im Byte D0 übertragen, Parameter des Typs Unsigned16 in D0 und D1.

Die Anzahl der gültigen Datenbytes ist im ersten CAN-Datenbyte (0x4x) wie folgt codiert:

Anzahl Parameter-Bytes	1	2	3	4
Erstes CAN-Datenbyte	0x4F	0x4B	0x47	0x43

Client -> Server, Download Request

11-bit Identifier	8 Byte Nutzdaten							
0x600 (=1536dez) + Node-ID	0x22	Index0	Index1	SubIdx	Data0	Data1	Data2	Data3

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)
Data0	Daten Low-Low-Byte (LLSB)
Data3	Daten High-High-Byte (MMSB)

Optional ist es möglich, im ersten CAN-Datenbyte die Anzahl der gültigen Parameter-Datenbytes anzugeben

Anzahl Parameter-Bytes	1	2	3	4
Erstes CAN-Datenbyte	0x2F	0x2B	0x27	0x23

In der Regel ist das jedoch nicht erforderlich, da jeweils nur die niederwertigen Datenbytes bis zur Länge des zu beschreibenden Objektverzeichniseintrags ausgewertet werden. Ein Download von Daten bis zu 4 Byte Länge kann daher bei BECKHOFF Busknoten immer mit 22 h im ersten CAN-Datenbyte erfolgen.

Client -> Server, Download Response

11-bit Identifier	8 Byte Nutzdaten							
0x580 (=1408dez) + Node-ID	0x60	Index0	Index1	SubIdx	0x00	0x00	0x00	0x00

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)

Abbruch Parameterkommunikation

Im Falle einer fehlerhaften Parameterkommunikation wird diese abgebrochen. Client bzw. Server senden dazu ein SDO-Telegramm folgender Struktur:

11-bit Identifier	8 Byte Nutzdaten							
0x580 (Client) oder 0x600(Server) + Node-ID	0x80	Index0	Index1	SubIdx	Error0	Error1	Error2	Error3

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)
Error0	SDO Fehler-Code Low-Low-Byte (LLSB)
Error3	SDO Fehler-Code High-High-Byte (MMSB)

Liste der SDO-Fehler-Codes (Abbruch-Grund des SDO-Transfers):

SDO Fehler-Code	Erläuterung
0x05 03 00 00	Toggle Bit nicht geändert
0x05 04 00 01	SDO Command Specifier ungültig oder unbekannt
0x06 01 00 00	Zugriff auf dieses Objekt wird nicht unterstützt
0x06 01 00 02	Versuch, auf einen Read_Only Parameter zu schreiben
0x06 02 00 00	Objekt nicht im Objektverzeichnis vorhanden
0x06 04 00 41	Objekt kann nicht ins PDO gemappt werden
0x06 04 00 42	Anzahl und/oder Länge der gemappten Objekte würde PDO Länge überschreiten
0x06 04 00 43	Allgemeine Parameter Inkompatibilität
0x06 04 00 47	Allgemeiner interner Fehler im Gerät
0x06 06 00 00	Zugriff wegen Hardware-Fehler abgebrochen
0x06 07 00 10	Datentyp oder Parameterlänge stimmen nicht überein oder sind unbekannt
0x06 07 00 12	Datentyp stimmt nicht überein, Parameterlänge zu groß
0x06 07 00 13	Datentyp stimmt nicht überein, Parameterlänge zu klein
0x06 09 00 11	Subindex nicht vorhanden
0x06 09 00 30	allgemeiner Wertebereich-Fehler
0x06 09 00 31	Wertebereich-Fehler: Parameter wert zu groß
0x06 09 00 32	Wertebereich-Fehler: Parameter wert zu klein
0x06 0A 00 23	Resource nicht verfügbar
0x08 00 00 21	Zugriff wegen lokaler Applikation nicht möglich
0x08 00 00 22	Zugriff wegen aktuellem Gerätestatus nicht möglich

Für die Register-Kommunikation (Index 0x4500, 0x4501) wurden weitere, herstellerspezifische Fehler-Codes eingeführt:

SDO Fehler-Code	Erläuterung
0x06 02 00 11	ungültige Tabelle: Tabelle oder Kanal nicht vorhanden
0x06 02 00 10	ungültiges Register: Tabelle nicht vorhanden
0x06 01 00 22	Schreibschutz noch gesetzt
0x06 07 00 43	fehlerhafte Anzahl Funktionsargumente
0x06 01 00 21	Funktion noch aktiv, später erneut versuchen
0x05 04 00 40	Allgemeiner Routing Fehler
0x06 06 00 21	Fehler Zugriff BC Tabelle
0x06 09 00 10	Allgemeiner Fehler bei Kommunikation mit Klemme
0x05 04 00 47	Time-out bei Kommunikation mit Klemme

5.6 SDO Kommunikation mit FC510x

CANopen SDO (Service Daten Objekt)-Kommunikation dient zum Auslesen bzw. Beschreiben beliebiger Parameter im Objektverzeichnis des CANopen Busknotens. Die FC5101CANopen PCI Karte benutzt die SDO Kommunikation zur Konfiguration der Kommunikationsparameter beim Aufstarten. Zusätzlich sind zwei Arten der anwendungsspezifischen SDO Kommunikation möglich:

1. Download von anwendungsspezifischen Parametern beim Aufstarten

Hierzu sind die entsprechenden Parameter im System Manager bei dem entsprechenden Knoten im Griff "SDO" einzugeben. In eckigen Klammern erscheinen die Objekte, die sich aus den Konfigurationen im Griff CAN Node ergeben. Anschließend können beliebige Objektverzeichniseinträge angefügt werden.

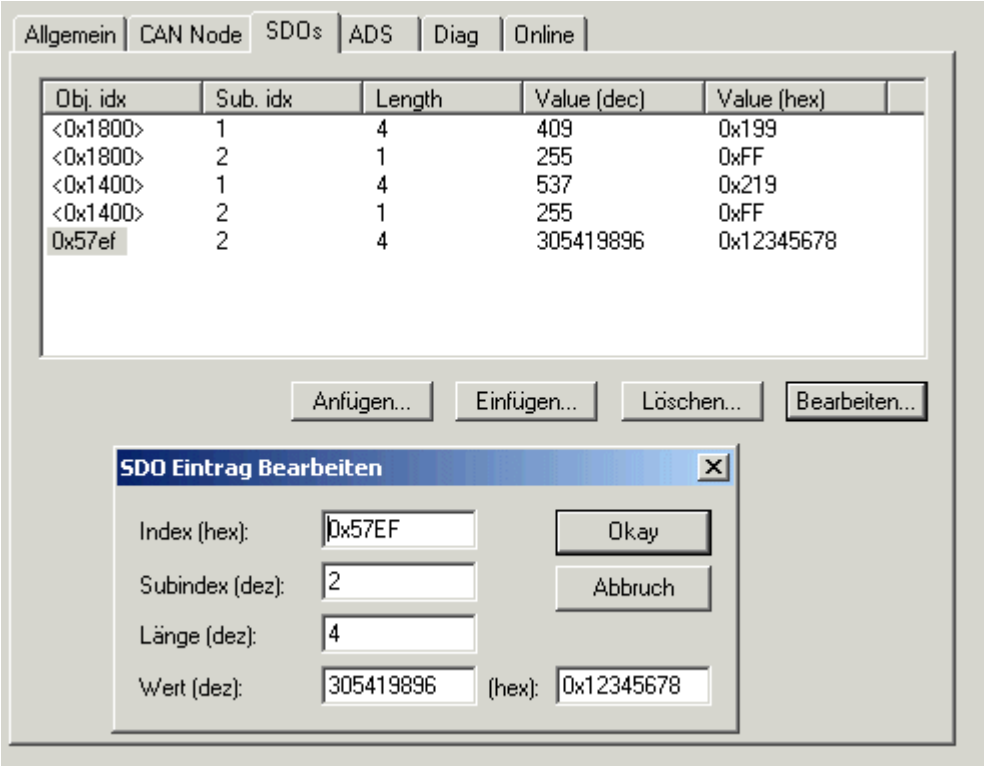


Abb. 44: SDO-Eintrag bearbeiten

Die Karte erwartet eine positive Quittierung des Parameterdownloads vom jeweiligen Busteilnehmer. Falls ein Parameter nicht geschrieben werden konnte (SDO-Abbruch durch Busteilnehmer), so versucht die Karte den entsprechenden Wert auszulesen und mit dem zu schreibenden Wert zu vergleichen - es könnte sich ja z. B. um einen Read_only Wert handeln, der bereits korrekt im Busteilnehmer konfiguriert ist. Bei Übereinstimmung geht die Karte zum nächsten Parametereintrag.

2. Upload und Download zur Laufzeit per ADS

Zur Laufzeit des Systems können SDO-Zugriffe auf die Objektverzeichnisse der Busteilnehmer über die Beckhoff ADS-Kommunikation erfolgen. Diese ist aus der SPS, aus der NC, vom OPC-Server, aus ActiveX-Controls oder von beliebigen anderen ADS Teilnehmern aus möglich.

Hierbei wird das SDO Protokoll komplett auf der Karte abgehandelt. Mit den ADS-Funktionen ADS Write bzw. ADS Read werden die Parameter auf die Karte übergeben und die Daten übergeben (Write) bzw. abgeholt (Read). Hierbei entspricht der Parameter "IDXGRP" dem 16 Bit-Index im CANopen Objektverzeichnis und "IDXOFFS" dem 8-Bit Subindex im CANopen Objektverzeichnis. Details zu den ADS Funktionsbausteinen finden sich in der TwinCAT Dokumentation (Beckhoff Information System).

Die Parameter der ADS Funktionsbausteine bilden sich wie folgt auf die SDO Parameter ab:

ADSREAD / ADSWRITE

Parameter	Beschreibung
NETID	Die NetID ist ein String mit 23 Byte Länge und ergibt sich per Default aus der IP-Adresse des Rechners, ergänzt um zwei Bytes. Sie adressiert die FC5101 Karte und kann dem Griff "ADS" im System Manager entnommen werden.
PORT	Enthält die Portnummer des ADS Gerätes - hier also die Portnummer des zu adressierenden CANopen Busteilnehmers.
IDXGRP	**Entspricht dem 16 Bit Index im CANopen Objektverzeichnis.**
IDXOFFS	**Entspricht dem 8 Bit Subindex im CANopen Objektverzeichnis.**
LEN	Die Länge des zu lesenden bzw. zu schreibenden Parameters in Bytes.
DESTADDR (nur ADSREAD)	Enthält die Adresse des Puffers, der die gelesenen Daten aufnehmen soll. Der Programmierer ist selbst dafür verantwortlich den Puffer in der Größe so zu dimensionieren, dass er ‚LEN' Bytes aufnehmen kann. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse man mit dem ADR - Operator ermitteln kann.
SRCADDR (nur ADSWRITE)	Enthält die Adresse des Puffers, aus dem die zu schreibenden Daten geholt werden sollen. Der Programmierer ist selbst dafür verantwortlich, den Puffer in der Größe so zu dimensionieren, dass ‚LEN'- Bytes daraus entnommen werden können. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse man mit dem ADR - Operator ermitteln kann.
READ	Durch eine steigende Flanke an diesem Eingang wird der ADS-Befehl ausgelöst
TIMEOUT	Gibt die Zeit bis zum Abbrechen der Funktion an
BUSY	Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt, längstens aber für die Dauer der, an dem 'Timeout'-Eingang angelegten, Zeit. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Bitte beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
ERR	Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung des Befehls ein Fehler aufgetreten ist.
ERRID	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das Ausführen eines Befehls an den Eingängen auf 0 zurückgesetzt.

Die ERRID ist ein 32 Bit Wert. Das Low-Word (Bits 0...15) enthält die allgemeinen ADS ERROR CODES, das High-Word (Bits 16...31) gibt SDO-spezifische Error Codes zurück:

MSB				LSB
Bit 31	Bit 30...24	Bit 23...20	Bit 19..16	Bit 15...0
1	Bit 6..0 des SDO Error Codes	Bit 19..16 des SDO Error Codes	Bit 27...24 des SDO Error Codes	ADS ERROR Code, Bedeutung siehe Kapitel Fehlerbehandlung und Diagnose [▶ 68]

Falls einer der Werte SDO Additional Code, SDO Error Code oder SDO Error Class größer ist als die zur Verfügung stehende Datenbreite (ausgeblendete Bits gesetzt), so wird im High-Word (Bits 16..31) der Wert 0x2115 zurückgegeben.

Beispiel: SDO Read per ADS

Im folgenden Beispielprogramm (Strukturierter Text) für die Verwendung der ADS Dienste für die SDO Kommunikation wird Objekt 0x1000, Subindex0 aus dem Knoten mit der Portnummer 0x1001 ausgelesen. Es handelt sich um den CANopen DeviceType. Dieser ist als UnSigned32 codiert und damit 4 Bytes lang.

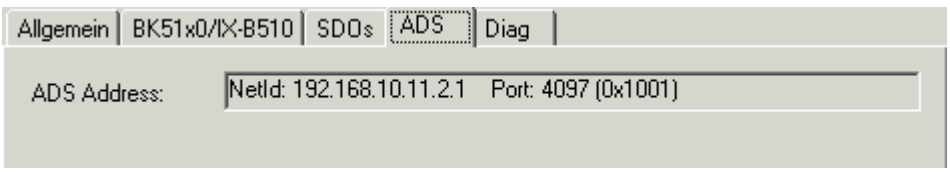


Abb. 45: SDO Read per ADS

```
SDO_READ(
        StartReading :=ReadStart,
        CO_Index :=16#1000,
        CO_SubIndex :=16#0,
        DataLength := 4,
        PortNr := 16#1001,
       ADSNetID:='192.168.10.11.2.1'
        );
IF SDO_READ.ReadDataAvailable THEN
        ReadStart :=FALSE;
        ReadError :=SDO_READ.Error;
        ReadData :=SDO_READ.ReadData;
END_IF
```

Der aufgerufene Funktionsbaustein SDO_READ ruft seinerseits mehrfach die ADSREAD Funktion auf. Er sieht wie folgt aus (zunächst die Variablendeklaration):

```
FUNCTION_BLOCK SDO_READ
VAR_INPUT
    ADSNetID:STRING(23); (* The AMSNetID addresses the FC5101 card. Can be empty if only one local s
ingle channel card is present*)
   PortNr:WORD;          (*This Port No. addresses the CANopen Node (see System Manager)*)
    CO_Index:DWORD;       (*This is the Index of the CANopen Object Dictionary Entry*)
    CO_SubIndex:DWORD;    (* This is the Sub-Index of the CANopen Object Dictionary Entry*)
    DataLength:DWORD;     (* This is the Length of the CANopen Object Dictionary Entry*)
    StartReading:BOOL;    (* only reset to FALSE after ReadDataAvailable=TRUE*)
END_VAR
VAR_OUTPUT
    ReadData:ARRAY[0..255] OF BYTE;
    ReadDataAvailable:BOOL;
    Error:DWORD;
END_VAR
VAR
    state:BYTE := 0;
    ADSREAD:ADSREAD;
END_VAR
```

```
CASE
state OF
    0:
        IF StartReading THEN
           ReadDataAvailable := FALSE;
           Error := 0;
           ADSRead(
               NETID:= ADSNetID,
               PORT:= PortNr,
               IDXGRP:= CO_Index,
               IDXOFFS:= CO_SubIndex,
               LEN:= DataLength,
               DESTADDR:= ADR(ReadData),
               READ:= TRUE,
               TMOUT := T#1s
               );
           IF ADSRead.err THEN
               state := 2;
               ReadDataAvailable := TRUE;
               Error := ADSRead.ErrId;
           ELSE
               state := 1;
           END_IF
        ELSE
           ADSRead(
               NETID:= ADSNetID,
               PORT:= PortNr,
               IDXGRP:= CO_Index,
               IDXOFFS:= CO_SubIndex,
               LEN:= DataLength,
               DESTADDR:= ADR(ReadData),
               READ:= FALSE,
               TMOUT := T#1s
           );
        END_IF
    1:
        ADSRead(READ:=FALSE);
        IF ADSRead.err THEN
           state := 2;
           ReadDataAvailable := TRUE;
           Error := ADSRead.ErrId;
        ELSE
```

```
            IF NOT ADSRead.busy THEN
                state := 2;
                ReadDataAvailable := TRUE;
            END_IF
        END_IF
    2:
        ADSRead(READ:=FALSE);
        state := 0;
END_CASE
```

Beispiel: SDO Write per ADS

Im folgenden Beispielprogramm (Strukturierter Text) für die Verwendung der ADS Dienste für die SDO Kommunikation wird Objekt 0x6200, Subindex3 aus dem Knoten mit der Portnummer 0x1001 beschrieben. Es handelt sich um digitale Ausgänge auf einem E/A Knoten.

```
(* Data to be written *)
WriteData[0] := 16#55;

(* write Object *)
    SDO_WRITE(
    StartWriting := WriteStart,
    CO_Index := 16#6200,
    CO_SubIndex := 3,
    DataLength := 1,
    PortNr := 16#1001,
    WriteData := WriteData,
    ADSNetID:='192.168.10.11.2.1'
    );
IF SDO_WRITE.WriteDataFinished THEN
    WriteStart := FALSE;
    WriteError := SDO_WRITE.Error;
END_IF
```

Der aufgerufene Funktionsbaustein SDO_WRITE ruft seinerseits mehrfach die ADSWRITE Funktion auf. Er sieht wie folgt aus (zunächst die Variablendeklaration):

```
FUNCTION_BLOCK SDO_WRITE
VAR_INPUT
    ADSNetID:STRING(23);   (* The AMSNetID addresses the FC5101 card. Can be empty if only one local
 single
channel card is present*)
   PortNr:WORD;          (* The Port No. addresses the CANopen Node (see System Manager)*)
   CO_Index:DWORD;        (* This is the Index of the CANopen Object Dictionary Entry*)
    CO_SubIndex:DWORD;     (*This is the Sub-Index of the CANopen Object Dictionary Entry*)
    DataLength:DWORD;     (* This is the Length of the CANopen Object Dictionary Entry*)
    StartWriting:BOOL;     (*only reset to FALSE after WriteDataFinished=TRUE*)
    WriteData:ARRAY[0..255] OF BYTE; (*This array contains the data to be written to the CANopen Obj
ect Dictionary*)
END_VAR
VAR_OUTPUT
    WriteDataFinished:BOOL;
    Error:DWORD;
END_VAR
VAR
    state:BYTE := 0;
    ADSWRITE:ADSWRITE;
END_VAR
```

```
CASE
state OF
    0:
        IF StartWriting THEN
           WriteDataFinished := FALSE;
           Error := 0;
           ADSWrite(
               NETID:= ADSNetID,
               PORT:= PortNr,
               IDXGRP:= CO_Index,
               IDXOFFS:= CO_SubIndex,
               LEN:= DataLength,
               SRCADDR:= ADR(WriteData),
               WRITE:= TRUE,
               TMOUT := T#1s
               );
```

```
            IF ADSWrite.err THEN
                state := 2;
                WriteDataFinished := TRUE;
                Error := ADSWrite.ErrId;
            ELSE
                state := 1;
            END_IF
        ELSE
            ADSWrite(
                NETID:= '',
                PORT:= PortNr,
                IDXGRP:= CO_Index,
                IDXOFFS:= CO_SubIndex,
                LEN:= DataLength,
                SRCADDR:= ADR(WriteData),
                WRITE:= FALSE,
                TMOUT := T#1s
                );
        END_IF
    1:
        ADSWrite(WRITE:=FALSE);
        IF ADSWrite.err THEN
            state := 2;
            WriteDataFinished := TRUE;
            Error := ADSWrite.ErrId;
        ELSE
            IF NOT ADSWrite.busy THEN
                state := 2;
                WriteDataFinished := TRUE;
            END_IF
        END_IF
    2:
        ADSWrite(WRITE:=FALSE);
        state := 0;
END_CASE
```

5.7 Baudrate und Bit Timing

Folgende Baudraten und Bittiming Register Einstellungen werden von den Beckhoff CANopen Geräten unterstützt:

Baudrate [kBaud]	BTR0	BTR1	Sampling Point
1000	0x00	0x14	75%
800	0x00	0x16	80%
500	0x00	0x1C	87%
250	0x01	0x1C	87%
125	0x03	0x1C	87%
100	0x04	0x1C	87%
50	0x09	0x1C	87%
20	0x18	0x1C	87%
10	0x31	0x1C	87%

Die angegebenen Bit-Timing Register Einstellungen (BTR0, BTR1) gelten z. B. für die CAN-Controller Philips 82C200, SJA1000, Intel 80C527, Siemens 80C167, und andere. Sie sind für maximale Buslänge optimiert.

5.8 Identifierverteilung

Default Identifier

CANopen sieht für die wichtigsten Kommunikationsobjekte Default-Identifier vor, die aus der 7-Bit Knotenadresse (Node-ID) und einem 4-Bit Function Code nach folgendem Schema abgeleitet werden:

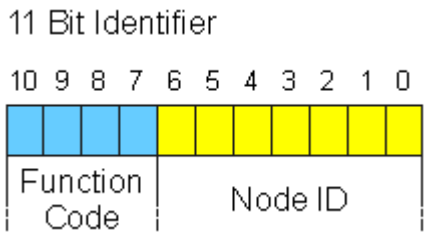


Abb. 46: Aufbau des Default Identifier

Für die Broadcast-Objekte wird die Node-ID *0* eingesetzt. Damit ergeben sich folgende Default-Identifier:

Broadcast-Objekte

Objekt	Funktion	Function Code	resultierende COB ID hex / dez	Objekt für Comm. Parameter / Mapping
NMT	Boot-Up	0	0x00 / 0	- / -
SYNC	Synchronisation	1	0x80 / 128	0x1005 + 0x1006 / -

Peer-to-Peer-Objekte

Objekt	Funktion bei E/A Geräten	Function Code	resultierende COB ID hex / dez	Objekt für Comm. Parameter / Mapping
Emergency	Status / Fehler	1	0x81 - 0xFF/ 129 - 255	- / -
PDO1 (tx)	dig. Eingänge	11	0x181 - 0x1FF / 385 - 511	0x1800 / 0x1A00
PDO1 (rx)	digitale Ausgänge	100	0x201 - 0x27F/ 513-639	0x1400 / 0x1600
PDO2 (tx)	analoge Eingänge	101	0x281 - 0x2FF/ 641-767	0x1801 / 0x1A01
PDO2 (rx)	analoge Ausgänge	110	0x301 - 0x37F/ 769-895	0x1401 / 0x1601
PDO3 (tx)	analoge Eingänge*	111	0x381 - 0x3FF / 897 - 1023	0x1802 / 0x1A02
PDO3 (rx)	analoge Ausgänge*	1000	0x401 - 0x47F/ 1025 - 1151	0x1402 / 0x1602
PDO4 (tx)	analoge Eingänge*	1001	0x481 - 0x4FF/ 1153 - 1279	0x1803 / 0x1A03
PDO4 (rx)	analoge Ausgänge*	1010	0x501 - 0x57F/ 1281 - 1407	0x1403 / 0x1603
SDO (tx)	Parameter	1011	0x581 - 0x5FF/ 1409-1535	- / -
SDO (rx)	Parameter	1100	0x601 - 0x67F/ 1537-1663	- / -
Guarding	Life-/Node-guarding, Heartbeat, Boot-Up Nachricht	1110	0x701 - 0x77F/ 1793-1919	(0x100C, 0x100D, 0x100E, 0x1016, 0x1017)

* Für PDO 3 + 4 gilt bei Beckhoff I/O Geräten aus historischen Gründen das Beckhoff Default Mapping. In den meisten Konfigurationen enthalten PDO 3+4 Daten von analogen Ein/Ausgängen, es können jedoch auch "überzählige" Daten von digitalen E/As oder Daten von Sonderklemmen sein. Details finden Sie in der Buskoppler Dokumentation.

Bis zur CANopen-Spezifikation Version 3 waren jeweils 2 PDOs mit Default-Identifiern versehen. Die Beckhoff Buskoppler bis Firmwarestand BA entsprechen diesem Stand der Spezifikation. Ab Firmwarestand C0 (CANopen Version 4) sind Default Identifier für bis zu 4 PDOs vorgesehen.

6 Fehlerbehandlung und Diagnose

6.1 LEDs

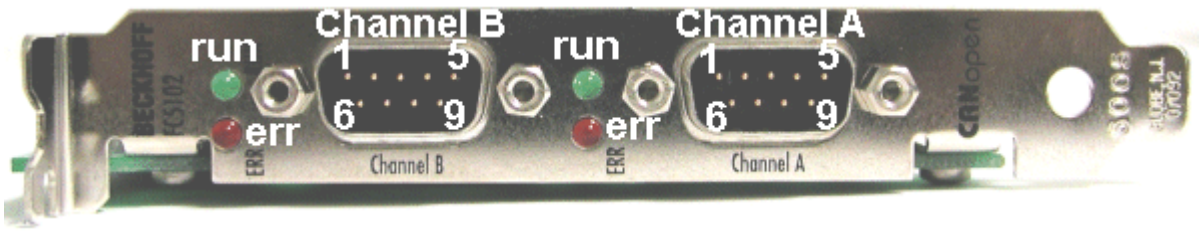


Abb. 47: FC510x - LEDs

LED Verhalten

Anhand der roten Error LED und der grünen Run LED lassen sich die wichtigsten Zustände der Karte schnell diagnostizieren:

Error LED (rot)	Run LED (grün)	Bedeutung
aus	aus	TwinCAT ist gestoppt
aus	an	Alle konfigurierten Busteilnehmer fehlerfrei (Box State=0), TwinCAT Task oder Prozess läuft.
aus	blinkt mit 2 Hz	Task, deren Prozessdaten mit der Karte verknüpft sind, läuft nicht. Alle konfigurierten Busteilnehmer gefunden und fehlerfrei (Box State=0)
blinkt mit 2 Hz	an	Mindestens ein Box State ist ungleich null (z. B. Teilnehmer nicht gefunden, falsche Konfiguration, Teilnehmer in Störung), TwinCAT Task läuft
blinkt mit 2 Hz	aus	Mindestens ein Box State ist ungleich null (z. B. Teilnehmer nicht gefunden, falsche Konfiguration, Teilnehmer in Störung), TwinCAT Task läuft nicht.
an	aus	TwinCAT läuft, CAN Controller ist Bus OFF. Physikalisches CAN Problem. Mögliche Fehlerursachen: z. B. fehlender Abschlusswiderstand, zu lange Busleitung, falsche Baudrate, Knotenadresse doppelt vergeben, Verdrahtungsfehler, Kurzschluss. Neustart erforderlich
blinkt mit 20 Hz	blinkt mit 20 Hz	Konfigurations-Upload wird durchgeführt
blinkt mit 20 Hz	aus	Karte ist in STOP Modus

6.2 Diagnose Busknoten

Die CANopen Feldbuskarte FC510x verfügt über umfangreiche Diagnosemöglichkeiten für die angeschlossenen Netzwerkknoten.

Für jeden CANopen Feldbusknoten gibt es die Eingangsvariable Node State, die den Status des jeweiligen Slaves zur Laufzeit signalisiert und z. B. mit der SPS verknüpft werden kann.

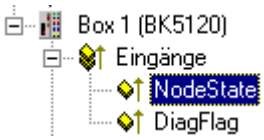


Abb. 48: Eingangsvariable Node State

Node-State (Box-State)

Node State	Bedeutung	Erläuterung
0 = 0x00	No error	Busknoten ist Operational, Kommunikation läuft fehlerfrei
1 = 0x01	Node deactivated	Knoten weist einen oder mehrere der folgenden Fehler auf: • Guarding/Heartbeat Fehler (Ausfall, Toggle Bit Fehler, Knoten hat Zustand gewechselt) • erwartetes TxPDO wurde nicht empfangen • TxPDO Länge kürzer als erwartet Knoten wurde gestoppt, da "Restart Manuell [▶ 31]" nach Knotenausfall angewählt wurde.
2 = 0x02	Node not found	Knoten wurde nicht gefunden: keine Antwort auf SDO-Lesezugriff auf Objekt 0x1000 an der erwarteten Knotenadresse. Am Knoten prüfen: eingestellte Knotenadresse + Baudrate. Netzwerk prüfen (Abschlusswiderstand, Stecker, Buslänge, vertauschte Leitungen, etc.)
4 = 0x04	SDO syntax error at StartUp	Fehler beim SDO Schreibzugriff: SDO Abort durch Knoten. Details siehe Karteireiter "Diag". oder: Länge eines via SDO gelesenen Objektes stimmt mit erwarteter Länge überein.
5 = 0x05	SDO data mismatch at StartUp	Erwartete Daten stimmen nicht mit via SDO gelesenen Daten überein (z. B. Device Profile und/oder Add. Info stimmen nicht mit Objekt 0x1000 überein). Tritt auch auf, wenn zu schreibender Wert (z. B. PDO COB-ID) wegen Verweigerung des Schreibzugriffs zurückgelesen wurde und nicht übereinstimmt. Details siehe Karteireiter "Diag".
8 = 0x08	Node StartUp in progress	Knoten wurde gefunden und wird gestartet.
11 = 0x0B	FC510x Bus-OFF	CAN Chip ist in "Bus-OFF" Zustand gegangen: Sende-Fehlerzähler ging
12 = 0x0C	Pre-Operational	Knoten ist (von selbst) in Pre-Operational gegangen.
13 = 0x0D	Severe bus fault	Allgemeiner Firmwarefehler.
14 = 0x0E	Guarding: toggle error	Guarding Fehler: Toggle Bit wurde nicht geändert.
20 = 0x14	TxPDO too short	Empfangenes TxPDO kürzer als erwartet.
22 = 0x16	Expected TxPDO is missing	TxPDO wurde im erwarteten Zeit-Intervall nicht empfangen : • Sync-Intervall bei synchronen TxPDOs, • Event-Timer bei ereignisgesteuerten PDOs.
23 = 0x17	Node is Operational but not all TxPDOs were received	Knoten wurde gestartet aber mindestens ein TxPDO des Knotens wurde noch nicht empfangen. Mögliche Ursachen (z. B.): • Knoten schickt ereignisgesteuerte PDOs erst nach dem ersten Event (nicht im Sinne der CANopen Spezifikation, aber durchaus üblich). • zu viele TxPDOs konfiguriert. • TxPDO ist auf dem Knoten vorhanden aber keine Prozessdaten gemappt. • TxPDO hat Transmission Type 1...120 (synchron), aber SYNC wurde noch nicht verschickt da zugehörige Task nicht gestartet wurde.

DiagFlag

Zeigt an, ob sich die Diagnoseinformationen der Box geändert haben.

Auslesen der Diagnosedaten via ADS

CANopen Emergencies und weitere Diagnosedaten können mit ADS-Read ausgelesen werden (neue Daten sind vorhanden, sobald das DiagFlag gesetzt ist). Dazu sind die ADS Net-ID der FC510x anzugeben. Weitere ADS Parameter:

Port: 200

IndexGroup: Lo-Word = 0xF180, Hi-Word = Node-Nummer.

IndexOffset: siehe unten

Länge: siehe unten

Wenn mehr als 26 Bytes Diagnosedaten ausgelesen werden, wird der Emergency-Speicher zurückgesetzt. Das DiagFlag wird zurückgesetzt, sobald ab Offset 0 mindestens 108 Bytes ausgelesen werden. Alternativ wird das Flag nach jedem Lesezugriff zurückgesetzt, wenn IndexGroup 0xF181 (statt 0xF180) zum Auslesen verwendet wird.

Die Diagnosedaten habe folgende Bedeutung:

Offset 0,1:	Bit 1:	Boot-Up-Message nicht empfangen oder fehlerhaft
	Bit 2:	Emergency-Overflow
	Bit 0, Bit 3-15:	reserviert
Offset 2,3:	Bit 0-14:	TX-PDO (i+1) empfangen
	Bit 15:	alle TX-PDOs 16-n empfangen
Offset 4,5:	Bit 0-4:	1. falsche TX-PDO-Länge 2. synchrone TX-PDO fehlt 3. Node meldet PRE-OPERATIONAL 4. Event-Timer bei einer TX-PDO abgelaufen 5. keine Antwort beim Guarden 6. mehrmals kein Toggeln beim Guarden
	Bit 5-15:	zugehörige COB-ID
Offset 6:	Bit 0-7:	1. falscher Wert bei einem SDO-Upload 2. falsche Länge bei einem SDO-Upload 3. Abort bei einem SDO-Up-/Download 4. falsches Datum bei einer Boot-Up-Message 5. Timeout beim Warten auf Boot-Up-Message
Offset 7:	Bit 0-7:	2: falscher SDO-Command specifier 3: SDO-Toggle-Bit hat sich nicht geändert 4: SDO-Länge zu groß 5: SDO-Abort 6: SDO-Timeout
Offset 8,9	Bit 0-7:	Index des SDO-Up/Downloads
Offset 10:	Bit 0-7:	Subindex des SDO-Up/Downloads
Offset 11:	Bit 0-7:	reserviert
Offset 12:	Bit 0-7:	errorClass des Aborts
Offset 13:	Bit 0-7:	errorCode des Aborts
Offset 14,15:	Bit 0-15:	additionalCode des Aborts
Offset 16-19:		gelesener Wert (falls Offset 6 = 1)
Offset 20-23:		erwarteter Wert (falls Offset 6 = 1)
Offset 24-25:		Anzahl der folgenden Emergencies
Offset 26 - n:		Emergencies (jeweils 8 Byte)

6.3 Diagnose FC510x

Die CANopen Feldbuskarte FC510x stellt umfangreiche Diagnosemöglichkeiten über Eingangsvariablen zur Verfügung.

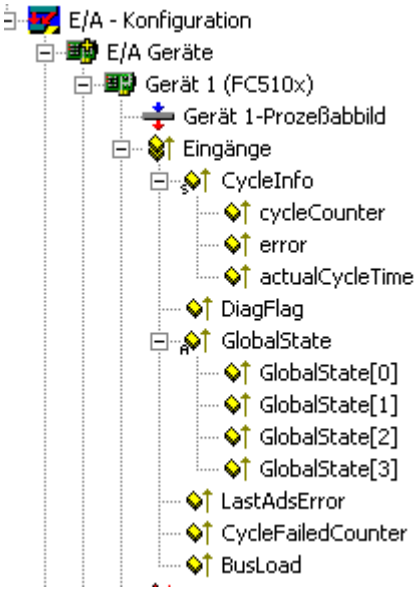


Abb. 49: FC510x - Diagnoseeingänge

cycleCounter

Wird nach jedem Firmware Zyklus inkrementiert. Kann dazu benutzt werden um von der SPS Task aus festzustellen, ob neue Eingangsdaten verarbeitet wurden - falls der cycleCounter seit dem letzten Aufruf der SPS Task nicht inkrementiert wurde, war die Task-Zeit zu kurz.

error

Anzahl der Knoten, deren Node State ungleich null ist.

actualCycleTime

Aktuelle Zyklus Zeit der Karten-Firmware in 4/25 µs. Abhängig von Datenaufkommen und Buslast.

DiagFlag

Wird auf 1 gesetzt wenn neue Diagnosedaten (z. B. Emergency) im Speicher der Karte angelegt wurden.

GlobalState

reserviert für interne Auswertungen.

LastAdsError

Letzter aufgetretener ADS Fehler. Siehe auch ADS Error Codes [▶ 68].

CycleFailedCounter

Dieser Zähler wird inkrementiert wenn der Firmware-Zyklus der Karte nicht abgeschlossen werden konnte bevor die höchstpriore verknüpfte Task erneutet auf das DPRAM zugreift. In diesem Fall erhält diese Task keine neuen Eingangsdaten, auch neue synchrone PDOs wurden im Zyklus davor nicht abgesetzt. Da der CycleFailedCounter erst *nach* dem entsprechenden Task-Start inkrementiert wird, kann er nicht zur Diagnose innerhalb dieser Task genutzt werden. Es empfiehlt sich, hierzu den cycleCounter heranzuziehen, der in diesem Fall nicht inkrementiert wurde.

Busload

Gibt die aktuelle Buslast in % an.

General Diag

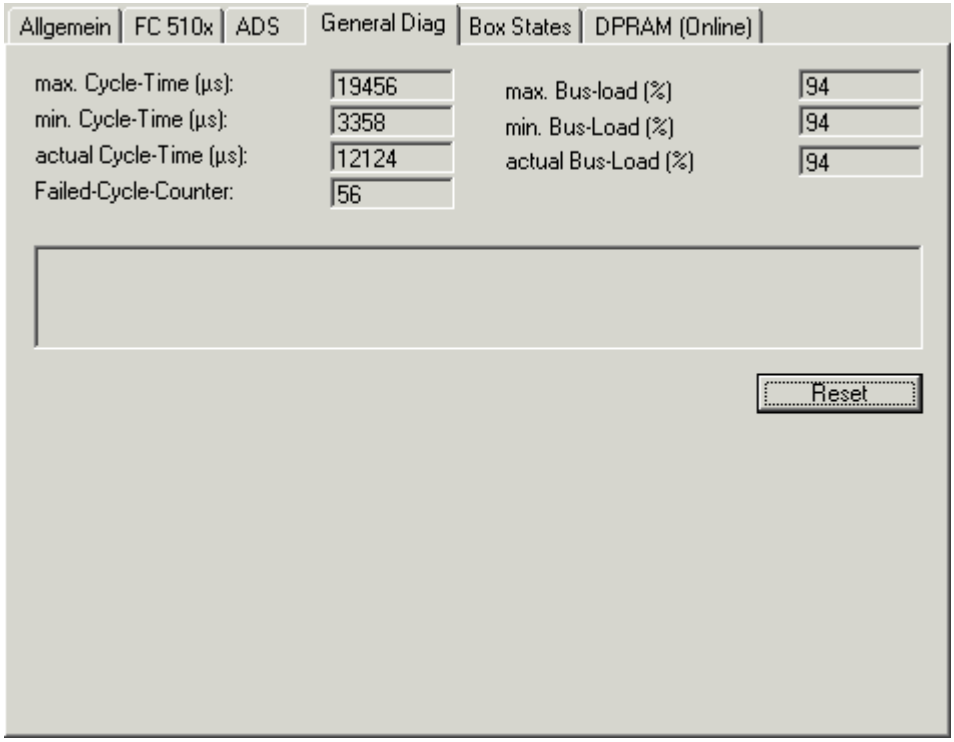


Abb. 50: Karteireiter "General Diag"

Im Karteireiter General Diag wird neben der aktuellen Buslast auch die minimale und maximale Buslast angezeigt - ebenso die Zykluszeit und der Failed Cycle Counter. Im oben gezeigten Beispiel werden ca. 5000 CAN-Frames/sec verarbeitet und entsprechend viele PDOs verschickt. Da die Firmware in jedem Zyklus alle anstehenden PDOs verschickt ergibt sich die Firmware Cycle Time in diesem Fall primär aus der Dauer, die für das Senden der PDOs benötigt wird.

6.4 Fehlertelegramme: Emergency

Die CANopen Feldbuskarte FC510x speichert eingehende Emergency Nachrichten im Diagnosebereich ab Offset 26 (siehe unten). Bis zu 10 Emergencies je Busknoten werden gespeichert. Wenn mehr Emergencies eintreffen wird die jeweils älteste Nachricht ersetzt.

Neue Diagnosedaten (Emergencies oder andere Diagnosedaten) sind vorhanden, sobald das DiagFlag gesetzt ist.

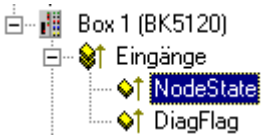


Abb. 51: Node State

CANopen Emergencies und weitere Diagnosedaten können mit ADS-Read ausgelesen werden. Dazu ist die ADS Net-ID der FC510x anzugeben. Weitere ADS Parameter:

Port: 200

IndexGroup: Lo-Word = 0xF180, Hi-Word = Node-Nummer.

IndexOffset: siehe unten

Länge: siehe unten

Wenn mehr als 26 Bytes Diagnosedaten ausgelesen werden, wird der Emergency-Speicher zurückgesetzt. Das DiagFlag wird zurückgesetzt, sobald ab Offset 0 mindestens 108 Bytes ausgelesen werden. Alternativ wird das Flag nach jedem Lesezugriff zurückgesetzt, wenn IndexGroup 0xF181 (statt 0xF180) zum Auslesen verwendet wird.

Die Beschreibung der Diagnosedaten an Offset 0...23 befindet sich im entsprechenden Kapitel [▶ 63]. Ab Offset 24 ist der Diagnosebereich wie folgt organisiert:

Offset 24-25:	Anzahl der folgenden Emergencies
Offset 26 - n:	Emergencies (jeweils 8 Byte)

Die Bedeutung der Emergency-Daten ist der technischen Dokumentation des jeweiligen CANopen Gerätes zu entnehmen.

6.5 ADS error codes

Die ADS Error Codes haben folgende Bedeutung:

Error	Beschreibung
0x1001	nicht genügend Speicher für AMS-Kommando
0x1101	falsche Datenlänge bei StartFieldbus
0x1102	falscher DeviceState bei StartFieldbus
0x1103	Device kann nicht von INIT nach RUN wechseln
0x1104	falscher AdsState im Zustand INIT
0x1105	falscher DeviceState bei StopFieldbus
0x1106	Device kann nicht STOP nach RUN wechseln, wenn keine CDL definiert ist
0x1107	Device kann nicht STOP nach RUN wechseln, wenn keine Box definiert ist
0x1108	falsche Datenlänge bei StartDataTransfer
0x1109	falscher DeviceState bei StartDataTransfer
0x110A	falscher AdsState im Zustand STOP
0x110B	Device kann nicht von RUN nach INIT wechseln
0x110C	falsche Datenlänge bei StopDataTransfer
0x110D	falscher DeviceState bei StopDataTransfer
0x1110	falscher AdsState im Zustand RUN
0x1111	Laden der Device-Parameter nur im Zustand INIT erlaubt
0x1112	falsche Datenlänge bei SetDeviceState
0x1113	AddBox im Zustand INIT nicht erlaubt
0x1114	falsche Datenlänge bei AddBox
0x1115	DeleteBox im Zustand INIT nicht erlaubt
0x1116	falscher IndexOffset bei DeleteBox
0x1117	falsche Datenlänge bei DeleteBox
0x1118	ReadBox nur mit AdsRead
0x1119	AddCdl im Zustand INIT nicht erlaubt
0x111A	falsche Datenlänge bei AddCdl
0x111B	DeleteCdl im Zustand INIT nicht erlaubt
0x111C	falscher IndexOffset bei DeleteCdl
0x111D	falsche Datenlänge bei DeleteCdl
0x111E	falsche IndexGroup bei AdsWrite
0x111F	Device-Parameter können nicht gelesen werden

Error	Beschreibung
0x1120	Box-Parameter können nicht gelesen werden
0x1121	Cdl-Parameter können nicht gelesen werden
0x1122	DeleteBox bzw. DeleteCdl nur mit AdsWrite
0x1123	ReadBox nur im Zustand STOP möglich
0x1124	falscher IndexOffset bei ReadBox
0x1125	falsche Datenlänge bei ReadBox
0x1126	falsche IndexGroup bei AdsRead
0x1127	AddDeviceNotification im Zustand INIT nicht erlaubt
0x1128	DelDeviceNotification im Zustand INIT nicht erlaubt
0x1129	IndexOffset zu groß beim Lesen der Device-Diagnosedaten
0x112B	IndexOffset zu groß beim Lesen der Box-Diagnosedaten
0x112F	nicht genügend Speicher für ReadBox-Response
0x113E	FC-Karte ist in einem Status in dem der Status Zustand nicht gewechselt werden kann
0x1201	AddCdl: Cdl-No ist zu groß
0x1202	DeleteCdl nur möglich, wenn CDL gestoppt ist
0x1203	DeleteCdl nicht möglich, da keine CDL definiert
0x1204	Zyklus konnte innerhalb der internen Watchdog-Zeit nicht beendet werden
0x1301	AddCdl: IO-Access-Multiplier ist zu groß
0x1302	AddCdl: Start-Cycle muss kleiner als IO-Access-Multiplier sein
0x1303	AddCdl: falsche Datenlänge der Output-Area
0x1304	AddCdl: falsche Datenoffset der Output-Area
0x1305	AddCdl: Output-Area ist bereits definiert
0x1306	AddCdl: falsche Datenlänge der Input-Area
0x1307	AddCdl: falsche Datenoffset der Input-Area
0x1308	AddCdl: Input-Area ist bereits definiert
0x1309	AddCdl: falscher Area-Typ
0x130A	AddCdl: BoxNo wurde nicht mit AddBox definiert
0x130B	AddCdl: falscher Aktions-Typ
0x130C	AddCdl: nicht genügend Speicher für Poll-Liste
0x130D	AddCdl: nicht genügend Speicher für Poll-Listen-Array
0x130E	AddCdl: nicht genügend Speicher für Aktionen
0x130F	AddCdl: CdlNo existiert bereits
0x1310	DeleteCdl: Cdl ist nicht gestoppt
0x1311	AddCdl: nicht genügend Speicher für asynchrone Sende-Liste
0x1312	AddCdl: nicht genügend Speicher für synchrone Empfangs-Liste
0x1313	AddCdl: nicht genügend Speicher für asynchrone Empfangs-Liste
0x1316	AddCdl: nicht genügend Speicher für synchrone Empfangs-Liste
0x1318	AddCdl: nur Slave-Aktion erlaubt
0x1319	AddCdl: nicht genügend Speicher für Slave-Liste

Error	Beschreibung
0x1601	AddBox: BoxNo ist zu groß
0x1602	AddBox: nicht genügend Speicher für ADS-StartUp-Telegramme
0x1604	DeleteBox: Box ist nicht gestoppt
0x1605	AddBox: nicht genügend Speicher für CDL-Telegramme
0x1606	AddBox: Anzahl der CDL-Telegramme ist zu groß
0x1607	BoxRestart: Box ist nicht gestoppt
0x1608	BoxRestart: Syntaxfehler AdsWriteControl
0x1609	BoxRestart: falscher AdsState
0x160A	Syntaxfehler bei AdsWrite an Box-Port
0x160B	AMS-CmdId wird von Box-Port nicht unterstützt
0x160E	AdsReadState wird von Box-Port nicht unterstützt
0x160F	AddBox: nicht genügend Speicher für das ADS-Interface
0x1610	AddBox: AMS-Channel ist ungültig
0x1611	Fehler Kommunikation zu einer AMS-Box
0x1613	Fehler Kommunikation zu einer AMS-Box: Offset ist falsch
0x1614	Fehler Kommunikation zu einer AMS-Box: Häppchen ist zu groß
0x1615	Fehler Kommunikation zu einer AMS-Box: AMS-Kommando ist zu groß
0x1616	Fehler Kommunikation zu einer AMS-Box: erstes Häppchen ist zu groß
0x1617	Fehler Kommunikation zu einer AMS-Box: erster Offset ist falsch
0x1701	AddDeviceNotification: Länge der Device-Diagnosedaten zu klein
0x1702	AddDeviceNotification: Länge der Device-Diagnosedaten zu groß
0x1703	AddDeviceNotification: Länge der Box-Diagnosedaten zu klein
0x1704	AddDeviceNotification: Länge der Box-Diagnosedaten zu groß
0x1705	AddDeviceNotification: Box ist nicht definiert
0x1706	AddDeviceNotification: falsche IndexGroup
0x1707	AddDeviceNotification: keine Resourcen mehr für Client
0x1708	DelDeviceNotification: falscher Handle
0x1801	StartFieldbus: Im Equidistant-Betrieb müssen Shift-Time + Safety-Time + 2*PLL-Sync-Time größer als die Cycle-Time sein
0x1802	StartFieldbus: Cycle-Time ist zu groß
0x1803	StartFieldbus: Cycle-Time ist zu groß
0x1804	StartFieldbus: Shift-Time ist zu groß
0x1805	StartFieldbus: PLL-Sync-Time ist zu groß
0x1806	StartFieldbus: Safety-Time ist zu groß
0x1807	StartFieldbus: Cycle-Times kleiner 1 ms müssen ein ganzzahliger Teiler von 1 ms sein

Error	Beschreibung
0x1A01	Speicher vom Huge-Heap konnte nicht alloziert werden, da er größer als 0x8000 Bytes war
0x1A02	Speicher vom Near-Heap konnte nicht alloziert werden, da er größer als 0x1000 Bytes war
0x1A03	Speicher vom Huge-Heap konnte nicht alloziert werden, da er 0 Bytes war
0x1A04	Speicher vom Near-Heap konnte nicht alloziert werden, da er 0 Bytes war
0x2001	StartFieldbus: Initialisierung des CAN-Controllers fehlgeschlagen
0x2002	AddBox: Box-Parameterlänge ist falsch
0x2003	AddBox: falsche Box-Nummer
0x2004	AddBox: Syntaxfehler bei ADS-StartUp-Parameter
0x2005	AddBox: Syntaxfehler bei PDO-Parameter
0x2006	AddBox: Syntaxfehler bei Datenlänge
0x2007	AddBox: nicht genügend Speicher
0x2008	AddCdl: falsche Empfangsdatenlänge
0x2009	AddCdl: falsche Sendedatenlänge
0x200A	AddCdl: PDO ist nicht definiert
0x200B	AddCdl: PDO-Id ist schon definiert
0x200C	AddBox: Syntaxfehler bei ADS-StartUp-Parameter
0x200D	AddBox: Syntaxfehler bei ADS-StartUp-Parameter
0x200E	AddBox: Emergency-Id ist schon definiert
0x200F	AddBox: zu viele PDOs definiert
0x2010	AddCdl: Telegrammindex ist falsch
0x2011	AddBox: zu viele Rx- bzw. Tx-PDOs
0x2012	AdsRead: falsche IndexGroup
0x2013	AdsRead: falscher IndexOffset
0x2014	AdsRead: falsche Länge
0x2015	AdsWrite: falsche IndexGroup
0x2016	AdsWrite: falscher IndexOffset
0x2017	AdsWrite: falsche Länge
0x2018	AddBox: Guarding-Time kleiner 10 ist nicht möglich
0x2019	AddBox: falscher Transmission-Type beim CAN-Layer 2-Node
0x201A	AdsRead: bei CAN-Layer 2-Node nicht möglich
0x201B	AdsWrite: bei CAN-Layer 2-Node nicht möglich
0x201C	AddBox: BootUp-Id ist schon definiert
0x201D	AddBox: BoxNo 0 ist nicht möglich
0x201E	StartFieldbus: Laden der Device-Device-Parameter nur im Zustand OFFLINE möglich
0x201F	StartDataTransfer: kein Speicher für Copy-Queue
0x2020	ReadBox: kein Speicher mehr
0x2021	ReadBox: SDO-Fehler oder Timeout
0x2022	ReadBox: SDO kann nicht initialisiert werden
0x2023	StartFieldbus: reserved Device-Parameter ungleich 0

Error	Beschreibung
0x2101	nicht genügend Speicher für niederpriore Queues
0x2102	nicht genügend Speicher für niederpriore Queues
0x2103	nicht genügend Speicher beim Node-Boot-Up
0x2104	nicht genügend Speicher beim Node-Boot-Up
0x2105	nicht genügend Speicher beim Node-Boot-Up
0x2106	nicht genügend Speicher beim Node-Boot-Up
0x2107	nicht genügend Speicher beim Node-Boot-Up
0x2108	nicht genügend Speicher beim Node-Boot-Up
0x2109	nicht genügend Speicher beim Node-Boot-Up
0x210A	nicht genügend Speicher beim Node-Boot-Up
0x210B	nicht genügend Speicher beim Node-Boot-Up
0x210C	nicht genügend Speicher beim Node-Boot-Up
0x210D	nicht genügend Speicher beim Node-Boot-Up
0x210E	nicht genügend Speicher beim Node-Boot-Up
0x210F	nicht genügend Speicher beim Node-Boot-Up
0x2110	nicht genügend Speicher beim Node-Boot-Up
0x2111	nicht genügend Speicher beim Node-Boot-Up
0x2112	nicht genügend Speicher beim Node-Boot-Up
0x2113	nicht genügend Speicher beim Node-Boot-Up
0x2114	nicht genügend Speicher beim Node-Boot-Up
0x2301	nicht genügend Speicher für niederpriore Queues
0x2302	nicht genügend Speicher für niederpriore Queues

6.6 CANopen Trouble Shooting

Error Frames

Fehler in der CAN-Verkabelung, der Adressvergabe und der Baud-Rateneinstellung zeigen sich u.a. durch eine erhöhte Anzahl an Error Frames: die Diagnose LEDs zeigen dann *Warning Limit wird überschritten* oder *Bus-Off-Zustand erreicht*.

Error Frames

Überschrittenes Warning Limit, Error Passive oder Bus-Off Zustand werden zunächst bei demjenigen Knoten angezeigt, der die meisten Fehler entdeckt hat. Dieser Knoten muss nicht unbedingt die Ursache für das Auftreten dieser Error Frames sein!
Wenn z. B. ein Knoten überdurchschnittlich stark zum Busverkehr beiträgt (z. B. weil er als einziger über analoge Eingänge verfügt, deren Daten in kurzen Abständen ereignisgesteuerte PDOs auslösen), so werden auch seine Telegramme mit großer Wahrscheinlichkeit zunächst gestört - entsprechend erreicht sein Fehlerzähler als erster kritische Zustände.

Node-ID / Baud Rate Einstellung

Es muss sorgfältig darauf geachtet werden, dass keine Knotenadresse doppelt vergeben ist: für jedes CAN-Datentelegramm darf es nur einen Sender geben.

Test 1

Knotenadressen überprüfen. Falls die CAN Kommunikation wenigstens zeitweise funktioniert und alle Geräte die Boot-Up-Nachricht unterstützen, so kann die Adressvergabe auch durch Aufzeichnen der Boot-Up-Nachrichten nach dem Einschalten der Geräte überprüft werden - hierdurch wird aber kein Vertauschen von Knotenadressen erkannt.

Test 2

Überprüfen, ob überall die gleiche Baud-Rate eingestellt ist. Bei Sondergeräten: Wenn Bittiming Parameter zugänglich, stimmen diese mit den CANopen-Definitionen überein (Abtastzeitpunkt, SJW, Oszillator).

Test der CAN-Verkabelung

Diese Tests nicht ausführen, wenn das Netzwerk aktiv ist: Während der Tests sollte keine Kommunikation stattfinden. Die folgenden Tests sollten in der angegebenen Reihenfolge ausgeführt werden, da manche Tests davon ausgehen, dass der vorhergehende Test erfolgreich war. In der Regel sind nicht alle Tests notwendig.

Netzwerkabschluss und Signalleitungen

Für diesen Test sollten die Knoten ausgeschaltet oder die CAN-Leitung abgesteckt sein, da die Messergebnisse sonst durch die aktiven CAN-Transceiver verfälscht werden können.

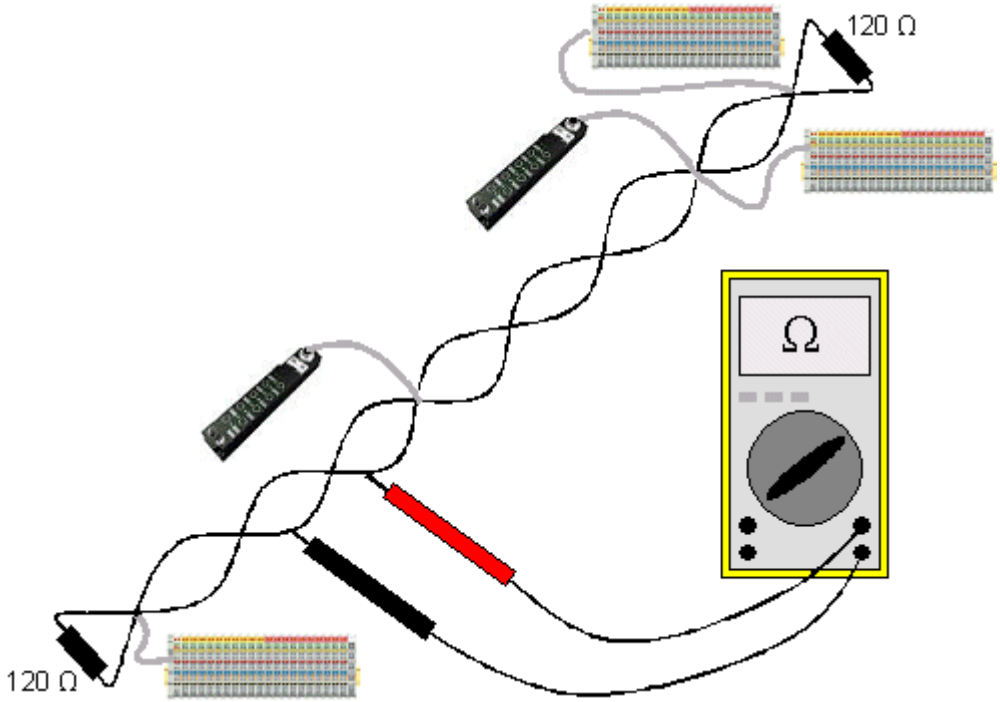


Abb. 52: Verdrahtungsplan für Testaufbau

Test 3

Widerstand zwischen CAN-high und CAN-low ermitteln - ggf. bei jedem Gerät.

Wenn der Messwert über 65 Ohm liegt, deutet dies auf fehlende Abschlusswiderstände oder den Bruch einer Signalleitung hin. Wenn der Messwert kleiner 50 Ohm ist, nach Kurzschluss zwischen CAN-Leitung, überzähligen Abschlusswiderständen oder fehlerhaften Transceivern suchen.

Test 4

Auf Kurzschluss zwischen CAN-Ground und den Signalleitungen sowie zwischen Schirm und Signalleitungen prüfen.

Test 5

Erdung von CAN-Ground und Schirm auftrennen. Auf Kurzschluss zwischen CAN-Ground und Schirm prüfen.

Topologie

Die Leitungslänge bei CAN Netzwerken hängt stark von der gewählten Baud-Rate ab. CAN toleriert dabei kurze Stichleitungen - ebenfalls in Abhängigkeit von der Baud-Rate. Die erlaubte Stichleitungslänge sollte nicht überschritten werden. Häufig wird die verlegte Leitungslänge unterschätzt - die Schätzung liegt teilweise Faktor 10 unter der tatsächlichen Länge. Deshalb empfiehlt sich folgender Test:

Test 6

Die Stichleitungslängen sowie die Busgesamtlänge nachmessen (nicht nur grob schätzen!) und mit den Topologieregeln (Baud-Ratenabhängig) vergleichen.

Schirmung und Erdung

Stromversorgung und Schirm sollten sorgfältig, einmalig und niederohmig beim Netzteil geerdet werden. Alle Verbindungsstellen, Abzweige etc. im CAN-Kabel müssen neben den Signalleitungen (und evtl. CAN-GND) auch den Schirm durchverbinden. In den Beckhoff IP20 Buskopplern wird der Schirm über ein R/C-Glied hochfrequenzmäßig geerdet.

Test 7

Mit DC-Strommessgerät (16 Amp max.) Strom zwischen Spannungsversorgungs-Masse und Schirm am vom Netzteil entfernten Ende des Netzes messen. Es sollte ein Ausgleichsstrom vorhanden sein. Wenn kein Strom vorhanden ist, so ist der Schirm nicht durchgängig verbunden oder das Netzteil ist nicht richtig geerdet. Wenn das Netzteil in der Mitte des Netzwerkes angeordnet ist, so sollte an beiden Enden gemessen werden. Dieser Test kann u.U. auch an den Stichleitungsenden durchgeführt werden.

Test 8

Den Schirm an mehreren Stellen auftrennen und den Verbindungsstrom messen. Wenn ein Stromfluss vorhanden ist, so ist der Schirm an mehreren Stellen geerdet (Erdschleife)

Potentialunterschiede

Der Schirm muss für diesen Test durchgängig sein und darf keinen Strom führen (vorher getestet).

Test 9

Spannung zwischen Schirm und Spannungsversorgungs-Erde an jedem Knoten ermitteln und notieren. Der maximale Potentialunterschied zwischen zwei beliebigen Geräten sollte kleiner als 5 Volt sein.

Fehler erkennen und lokalisieren

Am besten funktioniert in der Regel der "Low-tech-Ansatz": Teile des Netzes abhängen und beobachten, wann der Fehler verschwindet.

Aber: Dies funktioniert nicht gut bei Problemen wie zu großen Potentialunterschieden, Masseschleifen, EMV und Signalverfälschung da die Verkleinerung des Netzes häufig das Problem löst, ohne dass der „fehlende“ Teil ursächlich war. Auch die Buslast ändert sich beim Verkleinern des Netzes - damit können externe Störungen seltener CAN-Telegramme "treffen".

Die Diagnose mittels Oszilloskop führt meist nicht zum Erfolg: CAN Signale sehen auch im ungestörten Zustand teilweise recht wirr aus. Unter Umständen kann mit einem Speicheroszilloskop auf Error Frames getriggert werden - diese Art der Diagnose ist aber Messtechnik-Experten vorbehalten.

Protokollprobleme

In seltenen Fällen sind auch Protokollprobleme (z. B. fehlerhafte oder unvollständige CANopen-Implementierung, unglückliches Timing im Boot-Up etc.) Ursache von Störungen. Hier ist dann ein Mitschrieb (Trace) des Busverkehrs mit anschließender Auswertung durch CANopen Experten erforderlich - das Beckhoff Support Team kann hier helfen.
Für solch einen Trace eignet sich ein freier Kanal einer Beckhoff FC5102 CANopen PCI-Karte - die erforderliche Trace-Software stellt Beckhoff im Internet zur Verfügung. Alternativ kann selbstverständlich auch ein handelsübliches CAN Analysetool eingesetzt werden.

Protokollprobleme lassen sich vermeiden, indem auf den Einsatz von Geräten verzichtet wird, die nicht Conformance getestet sind. Der offizielle CANopen Conformance Test und das entsprechende Zertifikat sind beim CAN in Automation Verband (http://www.can-cia.de) erhältlich.

7 Bus Trace Funktion

7.1 FC510x als Bus Monitor

Die FC5101 bzw. FC5102 kann ab Firmware-Version 1.00 und TwinCAT 2.8 (Build 740) statt als Master bzw. Slave auch als CANopen-Monitor eingesetzt werden.

So kann z. B. der zweite Kanal der FC5102 für diesen Zweck genutzt werden, wobei der erste Kanal weiterhin als CANopen Master oder Slave fungiert, oder umgekehrt. In diesem Fall müssen beide Kanäle an das gleiche CAN Netzwerk angeschlossen werden (Der Datenaustausch innerhalb der Karte ist nicht vorgesehen, da dieser nicht rückwirkungsfrei erfolgen könnte).

Die von der FC510x aufgezeichneten Telegramme werden von der mit der FC510x verknüpften Task in einem Ring-Puffer zwischengespeichert, auf die gespeicherten Telegramme kann dann per ADS zugegriffen werden. Es gibt ein CANopen-Monitor-Programm (CANMON) mit Filter [▶ 77] und Trigger [▶ 77]-Möglichkeiten von Beckhoff als Free-Ware (siehe http://www.beckhoff.com im Download Bereich).

Einfügen FC510x als CANopen Monitor

Kontextmenü **Gerät anfügen**: CANopen Monitor einfügen

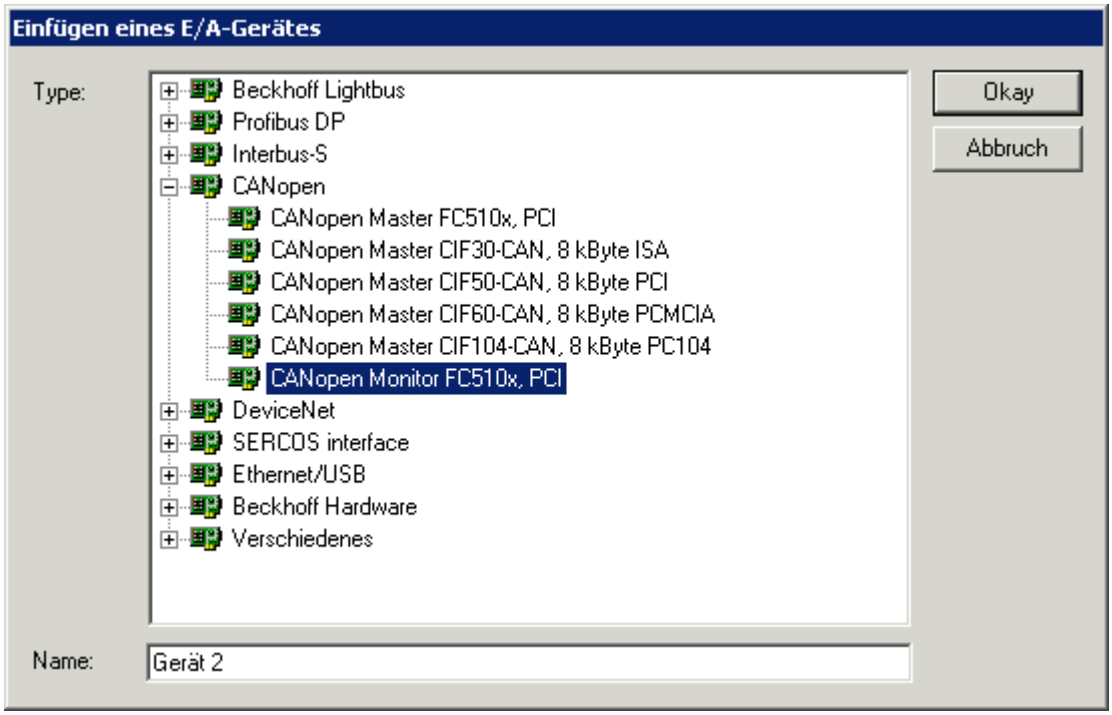


Abb. 53: Einfügen FC510x als CANopen Monitor

Danach muss der entsprechende Kanal (PCI-Speicheradresse) ausgewählt werden.

Verknüpfen der FC510x mit einer Task

Der Zugriff auf die Monitor-Daten erfolgt aus der TwinCAT-Echtzeit beim Start einer Task. Daher muss eine zusätzliche Task im System Manager angelegt werden, die mindestens eine UINT16-Eingangsvariable enthält, die mit einer der Variablen der FC510x zu verknüpfen ist.

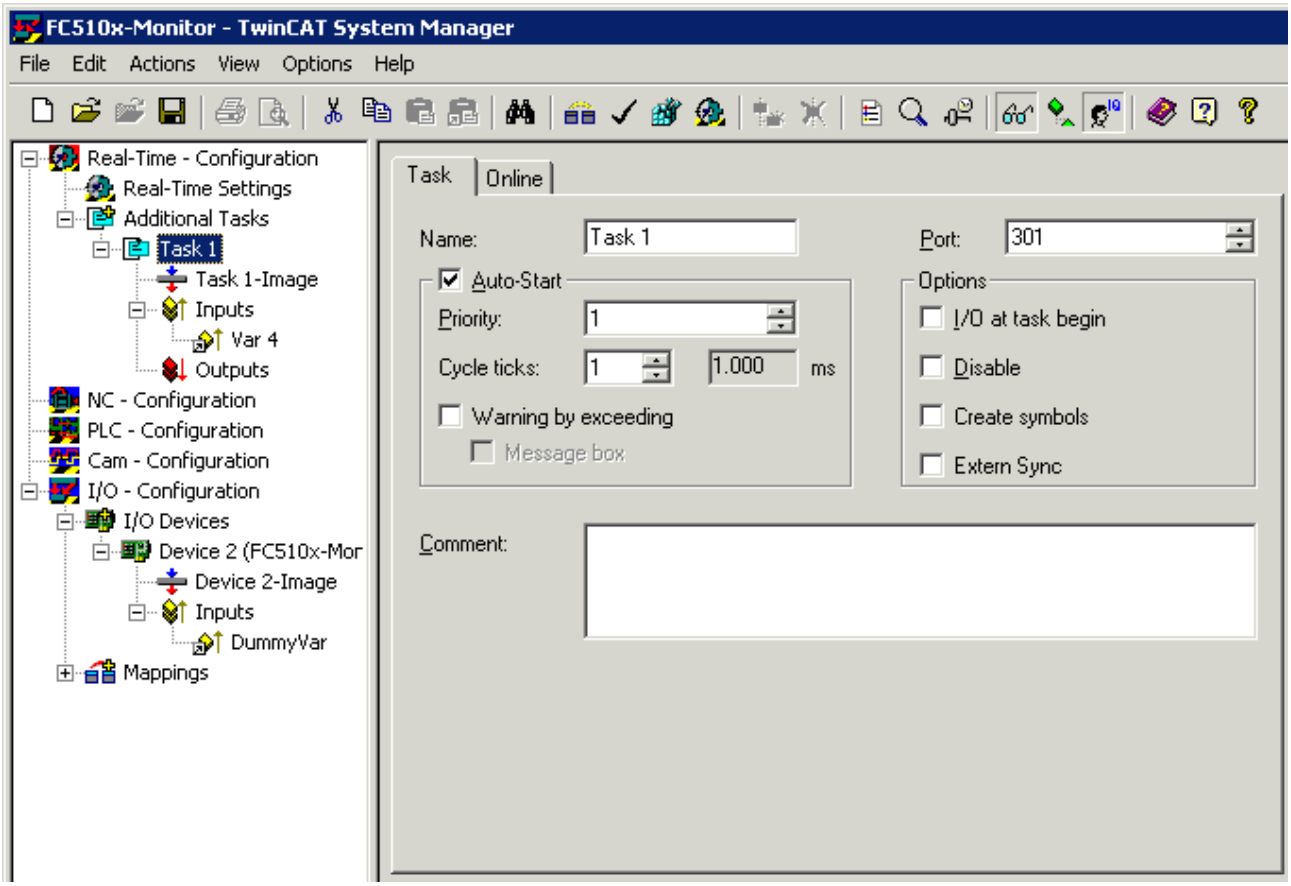


Abb. 54: Verknüpfen der FC510x mit einer Task

Die Verknüpfung dient nur dazu, dass aus der Echtzeit mit der Zykluszeit der Task auf die FC510x zugegriffen werden kann. Die Zykluszeit der zusätzlichen Task ist abhängig von der Baudrate wie folgt einzustellen:

Baudrate	Zykluszeit der zusätzlichen Task
1 MBaud	<= 10 ms
800 kBaud	<= 12 ms
500 kBaud	<= 20 ms
250 kBaud	<= 40 ms
125 kBaud	<= 80 ms
100 kBaud	<= 100 ms
50 kBaud	<= 200 ms
20 kBaud	<= 500 ms
10 kBaud	<= 1000 ms

Weiterhin ist die Check-Box Auto-Start zu setzen (s. Bild oben).

Karteireiter FCxxxx-Monitor

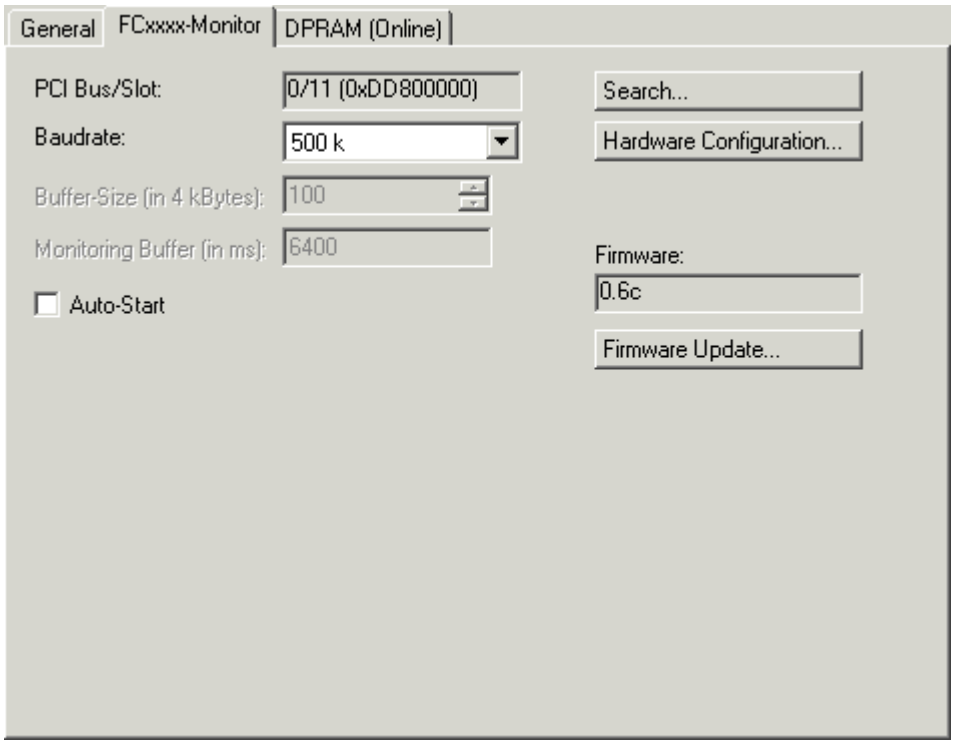


Abb. 55: Karteireiter FCxxxx-Monitor

PCI Slot/Irq: Zeigt an in welchem logischen PCI-Slot die Karte gefunden wurde.

Search...: Hierüber werden alle gesteckten FC510x-Kanäle gesucht, und es kann der gewünschte ausgewählt werden. Bei einer FC5102 erscheinen beide Kanäle A und B, die sich logisch wie zwei FC5101-Karten verhalten.

Hardware Configuration...: Hier kann die Hardware-Versionsnummer der FC510x angezeigt werden

Firmware: Hier wird die aktuelle Firmware-Version der FC510x angezeigt.

Firmware Update...: Hierüber kann die Firmware der FC510x-Karte aktualisiert werden.

Baudrate: Hier wird die CAN-Baudrate eingestellt.

Ring-Puffer: Hier wird die Größe des Ring-Puffers eingestellt, außerdem ist die Aufzeichnungsdauer für die eingestellte Ring-Puffer-Größe bei voller Buslastung angegeben.

Auto-Start: Wenn die Check-Box Auto-Start angeklickt ist, wird die Monitor-Aufzeichnung beim TwinCAT-Start gestartet, andernfalls muss der Start per ADS (via Monitor-Software) erfolgen.

Monitor Software

Die Monitor Software holt die Trace-Daten aus der FC510x Karte ab und legt sie als File auf dem gewünschten Massenspeicher ab. Zusätzlich zum eigentlichen Monitor-Programm (CAN-Monitor.exe) wird lediglich die DLL (TcRouterHelper.dll) benötigt. Diese hat im gleichen Verzeichnis wie das Monitor-Programm zu liegen.

Nach dem Start des Monitor-Programmes muss die Device-ID des FC5101-Kanals ausgewählt werden, der für das Monitoring verwendet wird.

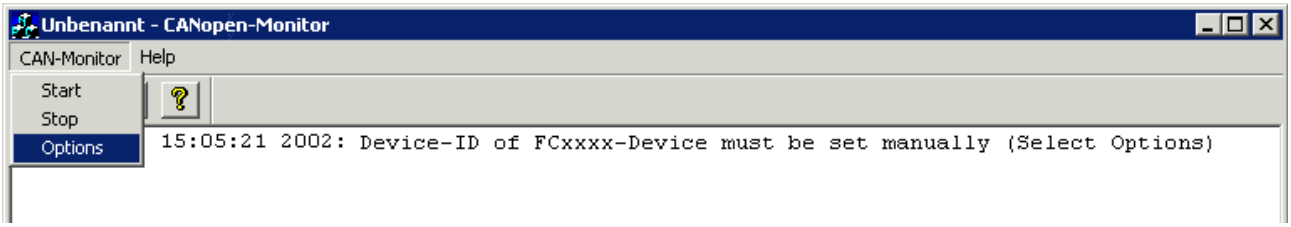


Abb. 56: CAN-Monitor - Aufruf der Optionen

Es öffnet sich das folgende Fenster:

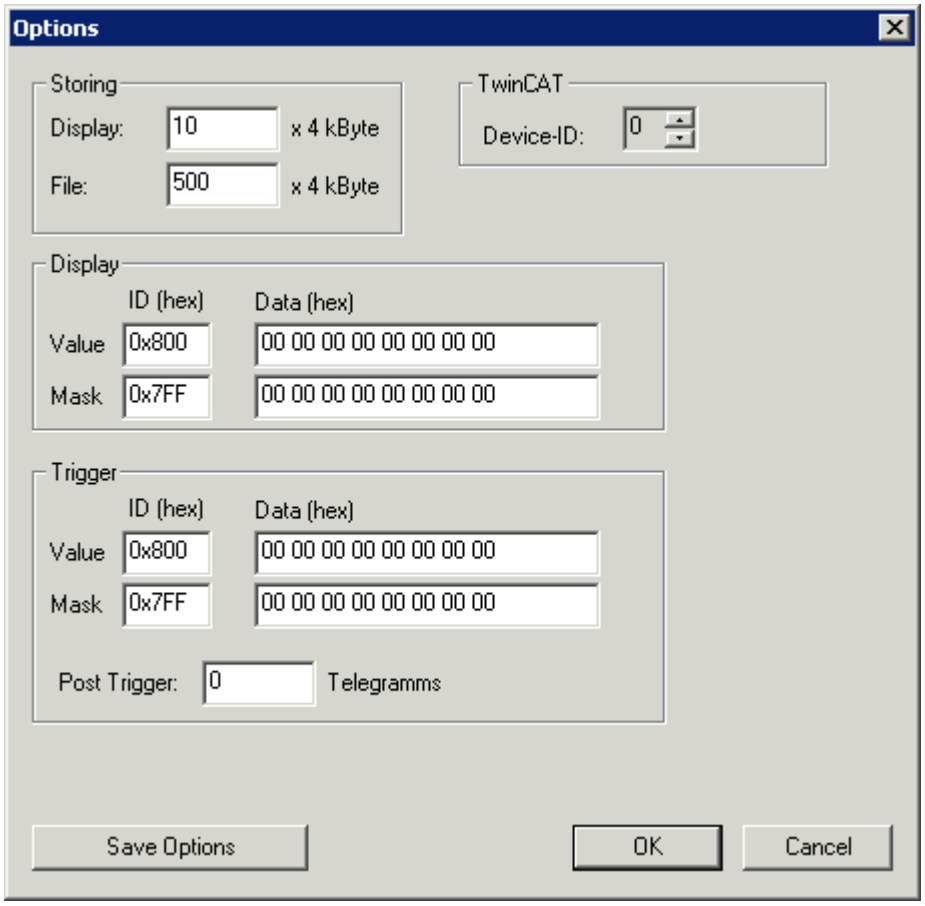


Abb. 57: CAN-Monitor - Optionen

Device-ID: Hier ist diejenige ID einzutragen, die vom System Manager dem FC510x-Monitor-Kanal zugeteilt wurde:

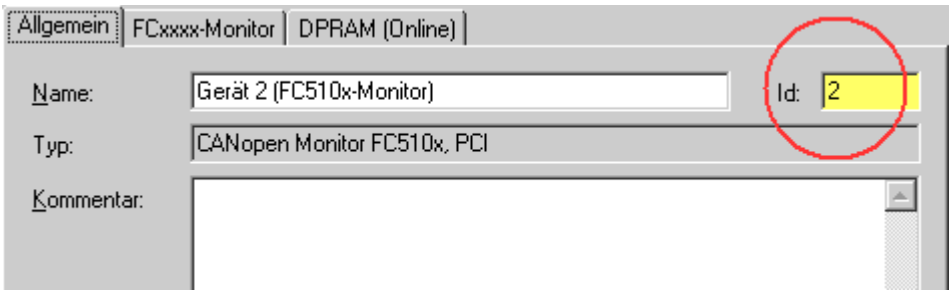


Abb. 58: Eintragen der Device-ID

Storing: Hier kann die Größe des Ringpuffer-Speichers für die Bildschirmausgabe (**Display**) und für die Dateiausgabe (**File**) eingestellt werden.

Display: Hier kann ein Filter für die Display-Ausgabe angegeben werden.

Trigger: Es können Trigger-Bedingungen angegeben werden, bei denen die Messung gestoppt werden soll. So kann beispielsweise bei Auftreten eines Fehlers ein bestimmter, sonst unbenutzter Ausgang gesetzt und auf dieses Bit im RxPDO der Ausgangsbaugruppe getriggert werden.

Post Trigger: Hier kann eingestellt werden, wie viele Telegramme nach Auftreten der Triggerbedingung noch aufgezeichnet werden sollen. Die Telegrammanzahl vor Auftreten der Triggerbedingung ist nur durch die eingestellte Pufferspeichergröße beschränkt.

Trigger und Display-Filter funktionieren so, dass bei der Auswahl der Identifier (ID) bzw. der Datenbytes (Data) nur diejenigen Bits berücksichtigt werden, bei denen die Maske 1 ist. Es werden dann diese Bits auf Übereinstimmung mit den in Value angegebenen Werten überprüft. Die Defaulteinstellung (Maske ID = 0x7FF und Maske Daten = 0x00) schlägt also zu, wenn der in Value ausgewählte Identifier (ID), unabhängig von den Daten, erkannt wird. Wenn bei Value ein Identifier (ID) größer 0x7FF (z. B. 0x800, Default) eingetragen wird, ist der Filter bzw. Triggerabgeschaltet (disabled).

Beispiel:
Die Messung soll nach 250 Telegrammen gestoppt werden, nachdem das Bit 0 im 2. Datenbyte des Telegramms mit Identifier 0x201gleich 1 ist. Hierzu ist folgendes einzutragen:

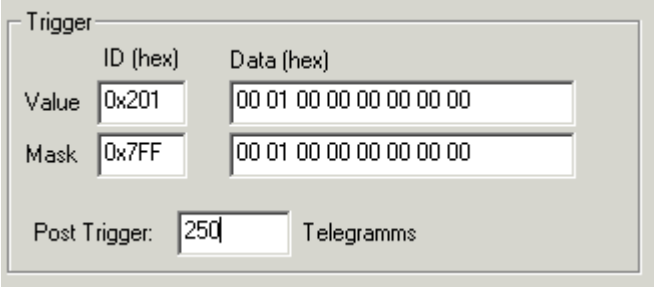


Abb. 59: Eintragen von 250 Telegrammen

Start der Aufzeichnung

TwinCAT muss gestartet sein, und eine Variable einer zyklischen Task (z. B. die Auto-Start-Task) muss mit der Dummy Variable der FC510x im Monitoring Mode verknüpft sein. Nun kann die Aufzeichnung durch Anklicken des grünen Ampel-Symbols gestartet werden.

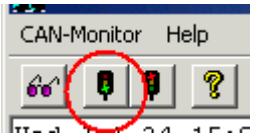


Abb. 60: Starten der Aufzeichnung

Falls der Aufstart-Vorgang eines weiteren CANopen Kanals im System aufgezeichnet werden soll, so ist die Check Box "Auto-Start" im Karteireiter FC-Monitor [▶ 77] des System Managers anzuwählen. Die Karte puffert dann bis zu 25000 CAN Nachrichten. Die Aufzeichnung via Monitor-Software muss dann lediglich gestartet werden, bevor der Puffer der CAN Karte überläuft.

Stopp der Aufzeichnung

Zum Beenden der Aufzeichnung ist das rote Ampel-Symbol anzuklicken. Es öffnet sich folgender Dialog:

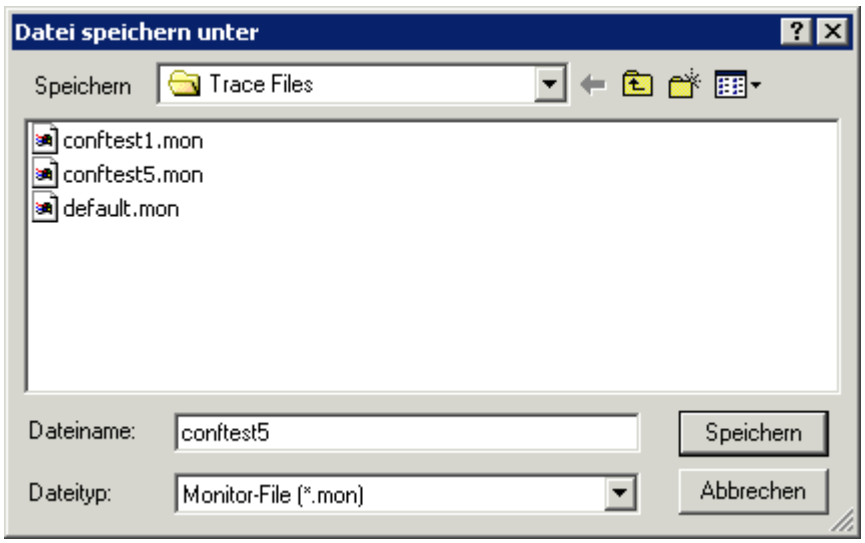


Abb. 61: Stoppen der Aufzeichnung

Es werden automatisch zwei Trace Files erzeugt: ein ASCII File mit der Endung *.mon, das mit jedem Text-Editor lesbar ist. Zusätzlich wird ein File mit der Endung *.ASC erzeugt, das mit dem CANalyzer® Tool der Firma Vector Informatik eingelesen und weiterverarbeitet werden kann.

Beispiel-Trace

Dargestellt ist der Beginn eines CANopen Boot-Ups mit zwei Knoten (Node ID 1 und Node ID 50).

Number Time(100 µs) Telegram
0 0.0638 Id: 000 Len: 2 Data: 82 00
1 0.0649 Id: 632 Len: 8 Data: 40 00 10 00 00 00 00 00
2 0.0653 Id: 601 Len: 8 Data: 40 00 10 00 00 00 00 00
3 2.0722 Id: 632 Len: 8 Data: 80 00 00 00 00 00 04 05
4 2.0725 Id: 601 Len: 8 Data: 80 00 00 00 00 00 04 05
5 2.2686 Id: 732 Len: 1 Data: 00
6 2.7440 Id: 701 Len: 1 Data: 00
7 4.0802 Id: 632 Len: 8 Data: 40 00 10 00 00 00 00 00
8 4.0806 Id: 601 Len: 8 Data: 40 00 10 00 00 00 00 00
9 4.0813 Id: 5b2 Len: 8 Data: 43 00 10 00 91 01 02 00
10 4.0823 Id: 632 Len: 8 Data: 40 18 10 01 00 00 00 00
11 4.0826 Id: 581 Len: 8 Data: 43 00 10 00 91 01 07 00
12 4.0835 Id: 601 Len: 8 Data: 40 18 10 01 00 00 00 00
13 4.0838 Id: 5b2 Len: 8 Data: 43 18 10 01 02 00 00 00
14 4.0848 Id: 632 Len: 8 Data: 23 00 14 01 32 02 00 00
15 4.0853 Id: 581 Len: 8 Data: 43 18 10 01 02 00 00 00
16 4.0863 Id: 601 Len: 8 Data: 23 00 18 01 81 01 00 00

8 Anhang

8.1 CAN Identifier-Liste

Die hier aufgeführte Liste soll bei der Identifizierung und Zuordnung von CANopen Nachrichten helfen. Aufgeführt sind alle von der CANopen Default Identifier Verteilung zugeordneten Identifier, sowie die von BECKHOFF via Objekt 0x5500 vergebenen herstellerspezifischen Default Identifier (nur in Netzen mit Knotenadressen <64 zu verwenden).

In der *chm-Ausgabe der Dokumentation dienen die folgenden Werte als Suchhilfe und "Einsprungpunkte" in die umfangreiche Identfier-Tabelle:

Dezimal: 400 [▶ 82], 500 [▶ 87], 600 [▶ 88], 700 [▶ 83], 800 [▶ 84], 900 [▶ 84], 1000 [▶ 89], 1100 [▶ 90], 1200 [▶ 85], 1300 [▶ 86], 1400 [▶ 91], 1500 [▶ 91], 1600 [▶ 92], 1700 [▶ 86], 1800 [▶ 94], 1900 [▶ 93]

Hexadezimal: 0x181 [▶ 82], 0x1C1 [▶ 87], 0x201 [▶ 83], 0x301 [▶ 84], 0x401 [▶ 85], 0x501 [▶ 86], 0x601 [▶ 94], 0x701 [▶ 94]

Die Identifier-Verteilung via Objekt 0x5500 folgt diesem Schema:

Objekt	resultierende COB-ID (dez)	resultierende COB-ID (hex)
Emergency [▶ 82]	129 bis 191 [255]	0x81 bis 0xBF [0xFF]
TxPDO1 [▶ 82]	385 bis 447 [511]	0x181 bis 0x1BF [0x1FF]
RxPDO1 [▶ 83]	513 bis 575 [639]	0x201 bis 0x23F [0x27F]
TxPDO2 [▶ 83]	641 bis 676 [767]	0x281 bis 0x2BF [0x2FF]
RxPDO2 [▶ 84]	769 bis 831 [895]	0x301 bis 0x33F [0x37F]
TxDPO3 [▶ 84]	897 bis 959 [1023]	0x381 bis 0x3BF [0x3FF]
RxPDO3 [▶ 85]	1025 bis 1087 [1151]	0x401 bis 0x43F [0x47F]
TxPDO4 [▶ 85]	1153 bis 1215 [1279]	0x481 bis 0x4BF [0x4FF]
RxPDO4 [▶ 86]	1281 bis 1343 [1407]	0x501 bis 0x53F [0x57F]
TxPDO5 [▶ 86]	1665 bis 1727	0x681 bis 0x6BF
RxPDO5 [▶ 87]	1921 bis 1983	0x781 bis 0x7BF
TxPDO6 [▶ 87]	449 bis 511	0x1C1 bis 0x1FF
RxPDO6 [▶ 88]	577 bis 639	0x241 bis 0x27F
TxDPO7 [▶ 88]	705 bis 767	0x2C1 bis 0x2FF
RxPDO7 [▶ 89]	833 bis 895	0x341 bis 0x37F
TxPDO8 [▶ 89]	961 bis 1023	0x3C1 bis 0x3FF
RxPDO8 [▶ 90]	1089 bis 1151	0x441 bis 0x47F
TxPDO9 [▶ 90]	1217 bis 1279	0x4C1 bis 0x4FF
RxPDO9 [▶ 91]	1345 bis 1407	0x541 bis 0x57F
TxDPO10 [▶ 91]	1473 bis 1535	0x5C1 bis 0x5FF
RxPDO10 [▶ 92]	1601 bis 1663	0x641 bis 0x67F
TxPDO11 [▶ 92]	1729 bis 1791	0x6C1 bis 0x6FF
RxPDO11 [▶ 93]	1857 bis 1919	0x741 bis 0x77F
SDO (Tx) [▶ 93]	1409 bis 1471 [1535]	0x581 bis 0x5BF [0x5FF]
SDO (Rx) [▶ 94]	1537 bis 1599 [1663]	0x601 bis 0x63F [0x67F]
Guarding / Heartbeat / Bootup [▶ 94]	1793 bis 1855 [1919]	0x701 bis 0x73F [0x77F]

Identifierliste

Mit * gekennzeichnete Identifier werden auf den Buskopplern nach Beschreiben von Index 0x5500 herstellerspezifisch vergeben.

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
0	0x00	NMT	149	0x95	EMCY Nd.21	171	0xAB	EMCY Nd.43
128	0x80	SYNC	150	0x96	EMCY Nd.22	172	0xAC	EMCY Nd.44
129	0x81	EMCY Nd.1	151	0x97	EMCY Nd.23	173	0xAD	EMCY Nd.45
130	0x82	EMCY Nd.2	152	0x98	EMCY Nd.24	174	0xAE	EMCY Nd.46
131	0x83	EMCY Nd.3	153	0x99	EMCY Nd.25	175	0xAF	EMCY Nd.47
132	0x84	EMCY Nd.4	154	0x9A	EMCY Nd.26	176	0xB0	EMCY Nd.48
133	0x85	EMCY Nd.5	155	0x9B	EMCY Nd.27	177	0xB1	EMCY Nd.49
134	0x86	EMCY Nd.6	156	0x9C	EMCY Nd.28	178	0xB2	EMCY Nd.50
135	0x87	EMCY Nd.7	157	0x9D	EMCY Nd.29	179	0xB3	EMCY Nd.51
136	0x88	EMCY Nd.8	158	0x9E	EMCY Nd.30	180	0xB4	EMCY Nd.52
137	0x89	EMCY Nd.9	159	0x9F	EMCY Nd.31	181	0xB5	EMCY Nd.53
138	0x8A	EMCY Nd.10	160	0xA0	EMCY Nd.32	182	0xB6	EMCY Nd.54
139	0x8B	EMCY Nd.11	161	0xA1	EMCY Nd.33	183	0xB7	EMCY Nd.55
140	0x8C	EMCY Nd.12	162	0xA2	EMCY Nd.34	184	0xB8	EMCY Nd.56
141	0x8D	EMCY Nd.13	163	0xA3	EMCY Nd.35	185	0xB9	EMCY Nd.57
142	0x8E	EMCY Nd.14	164	0xA4	EMCY Nd.36	186	0xBA	EMCY Nd.58
143	0x8F	EMCY Nd.15	165	0xA5	EMCY Nd.37	187	0xBB	EMCY Nd.59
144	0x90	EMCY Nd.16	166	0xA6	EMCY Nd.38	188	0xBC	EMCY Nd.60
145	0x91	EMCY Nd.17	167	0xA7	EMCY Nd.39	189	0xBD	EMCY Nd.61
146	0x92	EMCY Nd.18	168	0xA8	EMCY Nd.40	190	0xBE	EMCY Nd.62
147	0x93	EMCY Nd.19	169	0xA9	EMCY Nd.41	191	0xBF	EMCY Nd.63
148	0x94	EMCY Nd.20	170	0xAA	EMCY Nd.42			

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
385	0x181	TxPDO1, DI, Nd.1	406	0x196	TxPDO1, DI, Nd.22	427	0x1AB	TxPDO1, DI, Nd.43
386	0x182	TxPDO1, DI, Nd.2	407	0x197	TxPDO1, DI, Nd.23	428	0x1AC	TxPDO1, DI, Nd.44
387	0x183	TxPDO1, DI, Nd.3	408	0x198	TxPDO1, DI, Nd.24	429	0x1AD	TxPDO1, DI, Nd.45
388	0x184	TxPDO1, DI, Nd.4	409	0x199	TxPDO1, DI, Nd.25	430	0x1AE	TxPDO1, DI, Nd.46
389	0x185	TxPDO1, DI, Nd.5	410	0x19A	TxPDO1, DI, Nd.26	431	0x1AF	TxPDO1, DI, Nd.47
390	0x186	TxPDO1, DI, Nd.6	411	0x19B	TxPDO1, DI, Nd.27	432	0x1B0	TxPDO1, DI, Nd.48
391	0x187	TxPDO1, DI, Nd.7	412	0x19C	TxPDO1, DI, Nd.28	433	0x1B1	TxPDO1, DI, Nd.49
392	0x188	TxPDO1, DI, Nd.8	413	0x19D	TxPDO1, DI, Nd.29	434	0x1B2	TxPDO1, DI, Nd.50
393	0x189	TxPDO1, DI, Nd.9	414	0x19E	TxPDO1, DI, Nd.30	435	0x1B3	TxPDO1, DI, Nd.51
394	0x18A	TxPDO1, DI, Nd.10	415	0x19F	TxPDO1, DI, Nd.31	436	0x1B4	TxPDO1, DI, Nd.52
395	0x18B	TxPDO1, DI, Nd.11	416	0x1A0	TxPDO1, DI, Nd.32	437	0x1B5	TxPDO1, DI, Nd.53
396	0x18C	TxPDO1, DI, Nd.12	417	0x1A1	TxPDO1, DI, Nd.33	438	0x1B6	TxPDO1, DI, Nd.54
397	0x18D	TxPDO1, DI, Nd.13	418	0x1A2	TxPDO1, DI, Nd.34	439	0x1B7	TxPDO1, DI, Nd.55
398	0x18E	TxPDO1, DI, Nd.14	419	0x1A3	TxPDO1, DI, Nd.35	440	0x1B8	TxPDO1, DI, Nd.56
399	0x18F	TxPDO1, DI, Nd.15	420	0x1A4	TxPDO1, DI, Nd.36	441	0x1B9	TxPDO1, DI, Nd.57
400	0x190	TxPDO1, DI, Nd.16	421	0x1A5	TxPDO1, DI, Nd.37	442	0x1BA	TxPDO1, DI, Nd.58
401	0x191	TxPDO1, DI, Nd.17	422	0x1A6	TxPDO1, DI, Nd.38	443	0x1BB	TxPDO1, DI, Nd.59
402	0x192	TxPDO1, DI, Nd.18	423	0x1A7	TxPDO1, DI, Nd.39	444	0x1BC	TxPDO1, DI, Nd.60
403	0x193	TxPDO1, DI, Nd.19	424	0x1A8	TxPDO1, DI, Nd.40	445	0x1BD	TxPDO1, DI, Nd.61
404	0x194	TxPDO1, DI, Nd.20	425	0x1A9	TxPDO1, DI, Nd.41	446	0x1BE	TxPDO1, DI, Nd.62
405	0x195	TxPDO1, DI, Nd.21	426	0x1AA	TxPDO1, DI, Nd.42	447	0x1BF	TxPDO1, DI, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
513	0x201	RxPDO1, DO, Nd.1	534	0x216	RxPDO1, DO, Nd.22	555	0x22B	RxPDO1, DO, Nd.43
514	0x202	RxPDO1, DO, Nd.2	535	0x217	RxPDO1, DO, Nd.23	556	0x22C	RxPDO1, DO, Nd.44
515	0x203	RxPDO1, DO, Nd.3	536	0x218	RxPDO1, DO, Nd.24	557	0x22D	RxPDO1, DO, Nd.45
516	0x204	RxPDO1, DO, Nd.4	537	0x219	RxPDO1, DO, Nd.25	558	0x22E	RxPDO1, DO, Nd.46
517	0x205	RxPDO1, DO, Nd.5	538	0x21A	RxPDO1, DO, Nd.26	559	0x22F	RxPDO1, DO, Nd.47
518	0x206	RxPDO1, DO, Nd.6	539	0x21B	RxPDO1, DO, Nd.27	560	0x230	RxPDO1, DO, Nd.48
519	0x207	RxPDO1, DO, Nd.7	540	0x21C	RxPDO1, DO, Nd.28	561	0x231	RxPDO1, DO, Nd.49
520	0x208	RxPDO1, DO, Nd.8	541	0x21D	RxPDO1, DO, Nd.29	562	0x232	RxPDO1, DO, Nd.50
521	0x209	RxPDO1, DO, Nd.9	542	0x21E	RxPDO1, DO, Nd.30	563	0x233	RxPDO1, DO, Nd.51
522	0x20A	RxPDO1, DO, Nd.10	543	0x21F	RxPDO1, DO, Nd.31	564	0x234	RxPDO1, DO, Nd.52
523	0x20B	RxPDO1, DO, Nd.11	544	0x220	RxPDO1, DO, Nd.32	565	0x235	RxPDO1, DO, Nd.53
524	0x20C	RxPDO1, DO, Nd.12	545	0x221	RxPDO1, DO, Nd.33	566	0x236	RxPDO1, DO, Nd.54
525	0x20D	RxPDO1, DO, Nd.13	546	0x222	RxPDO1, DO, Nd.34	567	0x237	RxPDO1, DO, Nd.55
526	0x20E	RxPDO1, DO, Nd.14	547	0x223	RxPDO1, DO, Nd.35	568	0x238	RxPDO1, DO, Nd.56
527	0x20F	RxPDO1, DO, Nd.15	548	0x224	RxPDO1, DO, Nd.36	569	0x239	RxPDO1, DO, Nd.57
528	0x210	RxPDO1, DO, Nd.16	549	0x225	RxPDO1, DO, Nd.37	570	0x23A	RxPDO1, DO, Nd.58
529	0x211	RxPDO1, DO, Nd.17	550	0x226	RxPDO1, DO, Nd.38	571	0x23B	RxPDO1, DO, Nd.59
530	0x212	RxPDO1, DO, Nd.18	551	0x227	RxPDO1, DO, Nd.39	572	0x23C	RxPDO1, DO, Nd.60
531	0x213	RxPDO1, DO, Nd.19	552	0x228	RxPDO1, DO, Nd.40	573	0x23D	RxPDO1, DO, Nd.61
532	0x214	RxPDO1, DO, Nd.20	553	0x229	RxPDO1, DO, Nd.41	574	0x23E	RxPDO1, DO, Nd.62
533	0x215	RxPDO1, DO, Nd.21	554	0x22A	RxPDO1, DO, Nd.42	575	0x23F	RxPDO1, DO, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
641	0x281	TxPDO2, AI, Nd.1	662	0x296	TxPDO2, AI, Nd.22	683	0x2AB	TxPDO2, AI, Nd.43
642	0x282	TxPDO2, AI, Nd.2	663	0x297	TxPDO2, AI, Nd.23	684	0x2AC	TxPDO2, AI, Nd.44
643	0x283	TxPDO2, AI, Nd.3	664	0x298	TxPDO2, AI, Nd.24	685	0x2AD	TxPDO2, AI, Nd.45
644	0x284	TxPDO2, AI, Nd.4	665	0x299	TxPDO2, AI, Nd.25	686	0x2AE	TxPDO2, AI, Nd.46
645	0x285	TxPDO2, AI, Nd.5	666	0x29A	TxPDO2, AI, Nd.26	687	0x2AF	TxPDO2, AI, Nd.47
646	0x286	TxPDO2, AI, Nd.6	667	0x29B	TxPDO2, AI, Nd.27	688	0x2B0	TxPDO2, AI, Nd.48
647	0x287	TxPDO2, AI, Nd.7	668	0x29C	TxPDO2, AI, Nd.28	689	0x2B1	TxPDO2, AI, Nd.49
648	0x288	TxPDO2, AI, Nd.8	669	0x29D	TxPDO2, AI, Nd.29	690	0x2B2	TxPDO2, AI, Nd.50
649	0x289	TxPDO2, AI, Nd.9	670	0x29E	TxPDO2, AI, Nd.30	691	0x2B3	TxPDO2, AI, Nd.51
650	0x28A	TxPDO2, AI, Nd.10	671	0x29F	TxPDO2, AI, Nd.31	692	0x2B4	TxPDO2, AI, Nd.52
651	0x28B	TxPDO2, AI, Nd.11	672	0x2A0	TxPDO2, AI, Nd.32	693	0x2B5	TxPDO2, AI, Nd.53
652	0x28C	TxPDO2, AI, Nd.12	673	0x2A1	TxPDO2, AI, Nd.33	694	0x2B6	TxPDO2, AI, Nd.54
653	0x28D	TxPDO2, AI, Nd.13	674	0x2A2	TxPDO2, AI, Nd.34	695	0x2B7	TxPDO2, AI, Nd.55
654	0x28E	TxPDO2, AI, Nd.14	675	0x2A3	TxPDO2, AI, Nd.35	696	0x2B8	TxPDO2, AI, Nd.56
655	0x28F	TxPDO2, AI, Nd.15	676	0x2A4	TxPDO2, AI, Nd.36	697	0x2B9	TxPDO2, AI, Nd.57
656	0x290	TxPDO2, AI, Nd.16	677	0x2A5	TxPDO2, AI, Nd.37	698	0x2BA	TxPDO2, AI, Nd.58
657	0x291	TxPDO2, AI, Nd.17	678	0x2A6	TxPDO2, AI, Nd.38	699	0x2BB	TxPDO2, AI, Nd.59
658	0x292	TxPDO2, AI, Nd.18	679	0x2A7	TxPDO2, AI, Nd.39	700	0x2BC	TxPDO2, AI, Nd.60
659	0x293	TxPDO2, AI, Nd.19	680	0x2A8	TxPDO2, AI, Nd.40	701	0x2BD	TxPDO2, AI, Nd.61
660	0x294	TxPDO2, AI, Nd.20	681	0x2A9	TxPDO2, AI, Nd.41	702	0x2BE	TxPDO2, AI, Nd.62
661	0x295	TxPDO2, AI, Nd.21	682	0x2AA	TxPDO2, AI, Nd.42	703	0x2BF	TxPDO2, AI, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
769	0x301	RxPDO2, AO, Nd.1	790	0x316	RxPDO2, AO, Nd.22	811	0x32B	RxPDO2, AO, Nd.43
770	0x302	RxPDO2, AO, Nd.2	791	0x317	RxPDO2, AO, Nd.23	812	0x32C	RxPDO2, AO, Nd.44
771	0x303	RxPDO2, AO, Nd.3	792	0x318	RxPDO2, AO, Nd.24	813	0x32D	RxPDO2, AO, Nd.45
772	0x304	RxPDO2, AO, Nd.4	793	0x319	RxPDO2, AO, Nd.25	814	0x32E	RxPDO2, AO, Nd.46
773	0x305	RxPDO2, AO, Nd.5	794	0x31A	RxPDO2, AO, Nd.26	815	0x32F	RxPDO2, AO, Nd.47
774	0x306	RxPDO2, AO, Nd.6	795	0x31B	RxPDO2, AO, Nd.27	816	0x330	RxPDO2, AO, Nd.48
775	0x307	RxPDO2, AO, Nd.7	796	0x31C	RxPDO2, AO, Nd.28	817	0x331	RxPDO2, AO, Nd.49
776	0x308	RxPDO2, AO, Nd.8	797	0x31D	RxPDO2, AO, Nd.29	818	0x332	RxPDO2, AO, Nd.50
777	0x309	RxPDO2, AO, Nd.9	798	0x31E	RxPDO2, AO, Nd.30	819	0x333	RxPDO2, AO, Nd.51
778	0x30A	RxPDO2, AO, Nd.10	799	0x31F	RxPDO2, AO, Nd.31	820	0x334	RxPDO2, AO, Nd.52
779	0x30B	RxPDO2, AO, Nd.11	800	0x320	RxPDO2, AO, Nd.32	821	0x335	RxPDO2, AO, Nd.53
780	0x30C	RxPDO2, AO, Nd.12	801	0x321	RxPDO2, AO, Nd.33	822	0x336	RxPDO2, AO, Nd.54
781	0x30D	RxPDO2, AO, Nd.13	802	0x322	RxPDO2, AO, Nd.34	823	0x337	RxPDO2, AO, Nd.55
782	0x30E	RxPDO2, AO, Nd.14	803	0x323	RxPDO2, AO, Nd.35	824	0x338	RxPDO2, AO, Nd.56
783	0x30F	RxPDO2, AO, Nd.15	804	0x324	RxPDO2, AO, Nd.36	825	0x339	RxPDO2, AO, Nd.57
784	0x310	RxPDO2, AO, Nd.16	805	0x325	RxPDO2, AO, Nd.37	826	0x33A	RxPDO2, AO, Nd.58
785	0x311	RxPDO2, AO, Nd.17	806	0x326	RxPDO2, AO, Nd.38	827	0x33B	RxPDO2, AO, Nd.59
786	0x312	RxPDO2, AO, Nd.18	807	0x327	RxPDO2, AO, Nd.39	828	0x33C	RxPDO2, AO, Nd.60
787	0x313	RxPDO2, AO, Nd.19	808	0x328	RxPDO2, AO, Nd.40	829	0x33D	RxPDO2, AO, Nd.61
788	0x314	RxPDO2, AO, Nd.20	809	0x329	RxPDO2, AO, Nd.41	830	0x33E	RxPDO2, AO, Nd.62
789	0x315	RxPDO2, AO, Nd.21	810	0x32A	RxPDO2, AO, Nd.42	831	0x33F	RxPDO2, AO, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
897	0x381	TxPDO3*, Nd.1	918	0x396	TxPDO3*, Nd.22	939	0x3AB	TxPDO3*, Nd.43
898	0x382	TxPDO3*, Nd.2	919	0x397	TxPDO3*, Nd.23	940	0x3AC	TxPDO3*, Nd.44
899	0x383	TxPDO3*, Nd.3	920	0x398	TxPDO3*, Nd.24	941	0x3AD	TxPDO3*, Nd.45
900	0x384	TxPDO3*, Nd.4	921	0x399	TxPDO3*, Nd.25	942	0x3AE	TxPDO3*, Nd.46
901	0x385	TxPDO3*, Nd.5	922	0x39A	TxPDO3*, Nd.26	943	0x3AF	TxPDO3*, Nd.47
902	0x386	TxPDO3*, Nd.6	923	0x39B	TxPDO3*, Nd.27	944	0x3B0	TxPDO3*, Nd.48
903	0x387	TxPDO3*, Nd.7	924	0x39C	TxPDO3*, Nd.28	945	0x3B1	TxPDO3*, Nd.49
904	0x388	TxPDO3*, Nd.8	925	0x39D	TxPDO3*, Nd.29	946	0x3B2	TxPDO3*, Nd.50
905	0x389	TxPDO3*, Nd.9	926	0x39E	TxPDO3*, Nd.30	947	0x3B3	TxPDO3*, Nd.51
906	0x38A	TxPDO3*, Nd.10	927	0x39F	TxPDO3*, Nd.31	948	0x3B4	TxPDO3*, Nd.52
907	0x38B	TxPDO3*, Nd.11	928	0x3A0	TxPDO3*, Nd.32	949	0x3B5	TxPDO3*, Nd.53
908	0x38C	TxPDO3*, Nd.12	929	0x3A1	TxPDO3*, Nd.33	950	0x3B6	TxPDO3*, Nd.54
909	0x38D	TxPDO3*, Nd.13	930	0x3A2	TxPDO3*, Nd.34	951	0x3B7	TxPDO3*, Nd.55
910	0x38E	TxPDO3*, Nd.14	931	0x3A3	TxPDO3*, Nd.35	952	0x3B8	TxPDO3*, Nd.56
911	0x38F	TxPDO3*, Nd.15	932	0x3A4	TxPDO3*, Nd.36	953	0x3B9	TxPDO3*, Nd.57
912	0x390	TxPDO3*, Nd.16	933	0x3A5	TxPDO3*, Nd.37	954	0x3BA	TxPDO3*, Nd.58
913	0x391	TxPDO3*, Nd.17	934	0x3A6	TxPDO3*, Nd.38	955	0x3BB	TxPDO3*, Nd.59
914	0x392	TxPDO3*, Nd.18	935	0x3A7	TxPDO3*, Nd.39	956	0x3BC	TxPDO3*, Nd.60
915	0x393	TxPDO3*, Nd.19	936	0x3A8	TxPDO3*, Nd.40	957	0x3BD	TxPDO3*, Nd.61
916	0x394	TxPDO3*, Nd.20	937	0x3A9	TxPDO3*, Nd.41	958	0x3BE	TxPDO3*, Nd.62
917	0x395	TxPDO3*, Nd.21	938	0x3AA	TxPDO3*, Nd.42	959	0x3BF	TxPDO3*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1025	0x401	RxPDO3*, Nd.1	1046	0x416	RxPDO3*, Nd.22	1067	0x42B	RxPDO3*, Nd.43
1026	0x402	RxPDO3*, Nd.2	1047	0x417	RxPDO3*, Nd.23	1068	0x42C	RxPDO3*, Nd.44
1027	0x403	RxPDO3*, Nd.3	1048	0x418	RxPDO3*, Nd.24	1069	0x42D	RxPDO3*, Nd.45
1028	0x404	RxPDO3*, Nd.4	1049	0x419	RxPDO3*, Nd.25	1070	0x42E	RxPDO3*, Nd.46
1029	0x405	RxPDO3*, Nd.5	1050	0x41A	RxPDO3*, Nd.26	1071	0x42F	RxPDO3*, Nd.47
1030	0x406	RxPDO3*, Nd.6	1051	0x41B	RxPDO3*, Nd.27	1072	0x430	RxPDO3*, Nd.48
1031	0x407	RxPDO3*, Nd.7	1052	0x41C	RxPDO3*, Nd.28	1073	0x431	RxPDO3*, Nd.49
1032	0x408	RxPDO3*, Nd.8	1053	0x41D	RxPDO3*, Nd.29	1074	0x432	RxPDO3*, Nd.50
1033	0x409	RxPDO3*, Nd.9	1054	0x41E	RxPDO3*, Nd.30	1075	0x433	RxPDO3*, Nd.51
1034	0x40A	RxPDO3*, Nd.10	1055	0x41F	RxPDO3*, Nd.31	1076	0x434	RxPDO3*, Nd.52
1035	0x40B	RxPDO3*, Nd.11	1056	0x420	RxPDO3*, Nd.32	1077	0x435	RxPDO3*, Nd.53
1036	0x40C	RxPDO3*, Nd.12	1057	0x421	RxPDO3*, Nd.33	1078	0x436	RxPDO3*, Nd.54
1037	0x40D	RxPDO3*, Nd.13	1058	0x422	RxPDO3*, Nd.34	1079	0x437	RxPDO3*, Nd.55
1038	0x40E	RxPDO3*, Nd.14	1059	0x423	RxPDO3*, Nd.35	1080	0x438	RxPDO3*, Nd.56
1039	0x40F	RxPDO3*, Nd.15	1060	0x424	RxPDO3*, Nd.36	1081	0x439	RxPDO3*, Nd.57
1040	0x410	RxPDO3*, Nd.16	1061	0x425	RxPDO3*, Nd.37	1082	0x43A	RxPDO3*, Nd.58
1041	0x411	RxPDO3*, Nd.17	1062	0x426	RxPDO3*, Nd.38	1083	0x43B	RxPDO3*, Nd.59
1042	0x412	RxPDO3*, Nd.18	1063	0x427	RxPDO3*, Nd.39	1084	0x43C	RxPDO3*, Nd.60
1043	0x413	RxPDO3*, Nd.19	1064	0x428	RxPDO3*, Nd.40	1085	0x43D	RxPDO3*, Nd.61
1044	0x414	RxPDO3*, Nd.20	1065	0x429	RxPDO3*, Nd.41	1086	0x43E	RxPDO3*, Nd.62
1045	0x415	RxPDO3*, Nd.21	1066	0x42A	RxPDO3*, Nd.42	1087	0x43F	RxPDO3*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1153	0x481	TxPDO4*, Nd.1	1174	0x496	TxPDO4*, Nd.22	1195	0x4AB	TxPDO4*, Nd.43
1154	0x482	TxPDO4*, Nd.2	1175	0x497	TxPDO4*, Nd.23	1196	0x4AC	TxPDO4*, Nd.44
1155	0x483	TxPDO4*, Nd.3	1176	0x498	TxPDO4*, Nd.24	1197	0x4AD	TxPDO4*, Nd.45
1156	0x484	TxPDO4*, Nd.4	1177	0x499	TxPDO4*, Nd.25	1198	0x4AE	TxPDO4*, Nd.46
1157	0x485	TxPDO4*, Nd.5	1178	0x49A	TxPDO4*, Nd.26	1199	0x4AF	TxPDO4*, Nd.47
1158	0x486	TxPDO4*, Nd.6	1179	0x49B	TxPDO4*, Nd.27	1200	0x4B0	TxPDO4*, Nd.48
1159	0x487	TxPDO4*, Nd.7	1180	0x49C	TxPDO4*, Nd.28	1201	0x4B1	TxPDO4*, Nd.49
1160	0x488	TxPDO4*, Nd.8	1181	0x49D	TxPDO4*, Nd.29	1202	0x4B2	TxPDO4*, Nd.50
1161	0x489	TxPDO4*, Nd.9	1182	0x49E	TxPDO4*, Nd.30	1203	0x4B3	TxPDO4*, Nd.51
1162	0x48A	TxPDO4*, Nd.10	1183	0x49F	TxPDO4*, Nd.31	1204	0x4B4	TxPDO4*, Nd.52
1163	0x48B	TxPDO4*, Nd.11	1184	0x4A0	TxPDO4*, Nd.32	1205	0x4B5	TxPDO4*, Nd.53
1164	0x48C	TxPDO4*, Nd.12	1185	0x4A1	TxPDO4*, Nd.33	1206	0x4B6	TxPDO4*, Nd.54
1165	0x48D	TxPDO4*, Nd.13	1186	0x4A2	TxPDO4*, Nd.34	1207	0x4B7	TxPDO4*, Nd.55
1166	0x48E	TxPDO4*, Nd.14	1187	0x4A3	TxPDO4*, Nd.35	1208	0x4B8	TxPDO4*, Nd.56
1167	0x48F	TxPDO4*, Nd.15	1188	0x4A4	TxPDO4*, Nd.36	1209	0x4B9	TxPDO4*, Nd.57
1168	0x490	TxPDO4*, Nd.16	1189	0x4A5	TxPDO4*, Nd.37	1210	0x4BA	TxPDO4*, Nd.58
1169	0x491	TxPDO4*, Nd.17	1190	0x4A6	TxPDO4*, Nd.48	1211	0x4BB	TxPDO4*, Nd.59
1170	0x492	TxPDO4*, Nd.18	1191	0x4A7	TxPDO4*, Nd.49	1212	0x4BC	TxPDO4*, Nd.60
1171	0x493	TxPDO4*, Nd.19	1192	0x4A8	TxPDO4*, Nd.40	1213	0x4BD	TxPDO4*, Nd.61
1172	0x494	TxPDO4*, Nd.20	1193	0x4A9	TxPDO4*, Nd.41	1214	0x4BE	TxPDO4*, Nd.62
1173	0x495	TxPDO4*, Nd.21	1194	0x4AA	TxPDO4*, Nd.42	1215	0x4BF	TxPDO4*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1281	0x501	RxPDO4*, Nd.1	1302	0x516	RxPDO4*, Nd.22	1323	0x52B	RxPDO4*, Nd.43
1282	0x502	RxPDO4*, Nd.2	1303	0x517	RxPDO4*, Nd.23	1324	0x52C	RxPDO4*, Nd.44
1283	0x503	RxPDO4*, Nd.3	1304	0x518	RxPDO4*, Nd.24	1325	0x52D	RxPDO4*, Nd.45
1284	0x504	RxPDO4*, Nd.4	1305	0x519	RxPDO4*, Nd.25	1326	0x52E	RxPDO4*, Nd.46
1285	0x505	RxPDO4*, Nd.5	1306	0x51A	RxPDO4*, Nd.26	1327	0x52F	RxPDO4*, Nd.47
1286	0x506	RxPDO4*, Nd.6	1307	0x51B	RxPDO4*, Nd.27	1328	0x530	RxPDO4*, Nd.48
1287	0x507	RxPDO4*, Nd.7	1308	0x51C	RxPDO4*, Nd.28	1329	0x531	RxPDO4*, Nd.49
1288	0x508	RxPDO4*, Nd.8	1309	0x51D	RxPDO4*, Nd.29	1330	0x532	RxPDO4*, Nd.50
1289	0x509	RxPDO4*, Nd.9	1310	0x51E	RxPDO4*, Nd.30	1331	0x533	RxPDO4*, Nd.51
1290	0x50A	RxPDO4*, Nd.10	1311	0x51F	RxPDO4*, Nd.31	1332	0x534	RxPDO4*, Nd.52
1291	0x50B	RxPDO4*, Nd.11	1312	0x520	RxPDO4*, Nd.32	1333	0x535	RxPDO4*, Nd.53
1292	0x50C	RxPDO4*, Nd.12	1313	0x521	RxPDO4*, Nd.33	1334	0x536	RxPDO4*, Nd.54
1293	0x50D	RxPDO4*, Nd.13	1314	0x522	RxPDO4*, Nd.34	1335	0x537	RxPDO4*, Nd.55
1294	0x50E	RxPDO4*, Nd.14	1315	0x523	RxPDO4*, Nd.35	1336	0x538	RxPDO4*, Nd.56
1295	0x50F	RxPDO4*, Nd.15	1316	0x524	RxPDO4*, Nd.36	1337	0x539	RxPDO4*, Nd.57
1296	0x510	RxPDO4*, Nd.16	1317	0x525	RxPDO4*, Nd.37	1338	0x53A	RxPDO4*, Nd.58
1297	0x511	RxPDO4*, Nd.17	1318	0x526	RxPDO4*, Nd.38	1339	0x53B	RxPDO4*, Nd.59
1298	0x512	RxPDO4*, Nd.18	1319	0x527	RxPDO4*, Nd.39	1340	0x53C	RxPDO4*, Nd.60
1299	0x513	RxPDO4*, Nd.19	1320	0x528	RxPDO4*, Nd.40	1341	0x53D	RxPDO4*, Nd.61
1300	0x514	RxPDO4*, Nd.20	1321	0x529	RxPDO4*, Nd.41	1342	0x53E	RxPDO4*, Nd.62
1301	0x515	RxPDO4*, Nd.21	1322	0x52A	RxPDO4*, Nd.42	1343	0x53F	RxPDO4*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1665	0x681	TxPDO5*, Nd.1	1686	0x696	TxPDO5*, Nd.22	1707	0x6AB	TxPDO5*, Nd.43
1666	0x682	TxPDO5*, Nd.2	1687	0x697	TxPDO5*, Nd.23	1708	0x6AC	TxPDO5*, Nd.44
1667	0x683	TxPDO5*, Nd.3	1688	0x698	TxPDO5*, Nd.24	1709	0x6AD	TxPDO5*, Nd.45
1668	0x684	TxPDO5*, Nd.4	1689	0x699	TxPDO5*, Nd.25	1710	0x6AE	TxPDO5*, Nd.46
1669	0x685	TxPDO5*, Nd.5	1690	0x69A	TxPDO5*, Nd.26	1711	0x6AF	TxPDO5*, Nd.47
1670	0x686	TxPDO5*, Nd.6	1691	0x69B	TxPDO5*, Nd.27	1712	0x6B0	TxPDO5*, Nd.48
1671	0x687	TxPDO5*, Nd.7	1692	0x69C	TxPDO5*, Nd.28	1713	0x6B1	TxPDO5*, Nd.49
1672	0x688	TxPDO5*, Nd.8	1693	0x69D	TxPDO5*, Nd.29	1714	0x6B2	TxPDO5*, Nd.50
1673	0x689	TxPDO5*, Nd.9	1694	0x69E	TxPDO5*, Nd.30	1715	0x6B3	TxPDO5*, Nd.51
1674	0x68A	TxPDO5*, Nd.10	1695	0x69F	TxPDO5*, Nd.31	1716	0x6B4	TxPDO5*, Nd.52
1675	0x68B	TxPDO5*, Nd.11	1696	0x6A0	TxPDO5*, Nd.32	1717	0x6B5	TxPDO5*, Nd.53
1676	0x68C	TxPDO5*, Nd.12	1697	0x6A1	TxPDO5*, Nd.33	1718	0x6B6	TxPDO5*, Nd.54
1677	0x68D	TxPDO5*, Nd.13	1698	0x6A2	TxPDO5*, Nd.34	1719	0x6B7	TxPDO5*, Nd.55
1678	0x68E	TxPDO5*, Nd.14	1699	0x6A3	TxPDO5*, Nd.35	1720	0x6B8	TxPDO5*, Nd.56
1679	0x68F	TxPDO5*, Nd.15	1700	0x6A4	TxPDO5*, Nd.36	1721	0x6B9	TxPDO5*, Nd.57
1680	0x690	TxPDO5*, Nd.16	1701	0x6A5	TxPDO5*, Nd.37	1722	0x6BA	TxPDO5*, Nd.58
1681	0x691	TxPDO5*, Nd.17	1702	0x6A6	TxPDO5*, Nd.38	1723	0x6BB	TxPDO5*, Nd.59
1682	0x692	TxPDO5*, Nd.18	1703	0x6A7	TxPDO5*, Nd.39	1724	0x6BC	TxPDO5*, Nd.60
1683	0x693	TxPDO5*, Nd.19	1704	0x6A8	TxPDO5*, Nd.40	1725	0x6BD	TxPDO5*, Nd.61
1684	0x694	TxPDO5*, Nd.20	1705	0x6A9	TxPDO5*, Nd.41	1726	0x6BE	TxPDO5*, Nd.62
1685	0x695	TxPDO5*, Nd.21	1706	0x6AA	TxPDO5*, Nd.42	1727	0x6BF	TxPDO5*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1921	0x781	RxPDO5*, Nd.1	1942	0x796	RxPDO5*, Nd.22	1963	0x7AB	RxPDO5*, Nd.43
1922	0x782	RxPDO5*, Nd.2	1943	0x797	RxPDO5*, Nd.23	1964	0x7AC	RxPDO5*, Nd.44
1923	0x783	RxPDO5*, Nd.3	1944	0x798	RxPDO5*, Nd.24	1965	0x7AD	RxPDO5*, Nd.45
1924	0x784	RxPDO5*, Nd.4	1945	0x799	RxPDO5*, Nd.25	1966	0x7AE	RxPDO5*, Nd.46
1925	0x785	RxPDO5*, Nd.5	1946	0x79A	RxPDO5*, Nd.26	1967	0x7AF	RxPDO5*, Nd.47
1926	0x786	RxPDO5*, Nd.6	1947	0x79B	RxPDO5*, Nd.27	1968	0x7B0	RxPDO5*, Nd.48
1927	0x787	RxPDO5*, Nd.7	1948	0x79C	RxPDO5*, Nd.28	1969	0x7B1	RxPDO5*, Nd.49
1928	0x788	RxPDO5*, Nd.8	1949	0x79D	RxPDO5*, Nd.29	1970	0x7B2	RxPDO5*, Nd.50
1929	0x789	RxPDO5*, Nd.9	1950	0x79E	RxPDO5*, Nd.30	1971	0x7B3	RxPDO5*, Nd.51
1930	0x78A	RxPDO5*, Nd.10	1951	0x79F	RxPDO5*, Nd.31	1972	0x7B4	RxPDO5*, Nd.52
1931	0x78B	RxPDO5*, Nd.11	1952	0x7A0	RxPDO5*, Nd.32	1973	0x7B5	RxPDO5*, Nd.53
1932	0x78C	RxPDO5*, Nd.12	1953	0x7A1	RxPDO5*, Nd.33	1974	0x7B6	RxPDO5*, Nd.54
1933	0x78D	RxPDO5*, Nd.13	1954	0x7A2	RxPDO5*, Nd.34	1975	0x7B7	RxPDO5*, Nd.55
1934	0x78E	RxPDO5*, Nd.14	1955	0x7A3	RxPDO5*, Nd.35	1976	0x7B8	RxPDO5*, Nd.56
1935	0x78F	RxPDO5*, Nd.15	1956	0x7A4	RxPDO5*, Nd.36	1977	0x7B9	RxPDO5*, Nd.57
1936	0x790	RxPDO5*, Nd.16	1957	0x7A5	RxPDO5*, Nd.37	1978	0x7BA	RxPDO5*, Nd.58
1937	0x791	RxPDO5*, Nd.17	1958	0x7A6	RxPDO5*, Nd.38	1979	0x7BB	RxPDO5*, Nd.59
1938	0x792	RxPDO5*, Nd.18	1959	0x7A7	RxPDO5*, Nd.39	1980	0x7BC	RxPDO5*, Nd.60
1939	0x793	RxPDO5*, Nd.19	1960	0x7A8	RxPDO5*, Nd.40	1981	0x7BD	RxPDO5*, Nd.61
1940	0x794	RxPDO5*, Nd.20	1961	0x7A9	RxPDO5*, Nd.41	1982	0x7BE	RxPDO5*, Nd.62
1941	0x795	RxPDO5*, Nd.21	1962	0x7AA	RxPDO5*, Nd.42	1983	0x7BF	RxPDO5*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
449	0x1C1	TxPDO6*, Nd.1	470	0x1D6	TxPDO6*, Nd.22	491	0x1EB	TxPDO6*, Nd.43
450	0x1C2	TxPDO6*, Nd.2	471	0x1D7	TxPDO6*, Nd.23	492	0x1EC	TxPDO6*, Nd.44
451	0x1C3	TxPDO6*, Nd.3	472	0x1D8	TxPDO6*, Nd.24	493	0x1ED	TxPDO6*, Nd.45
452	0x1C4	TxPDO6*, Nd.4	473	0x1D9	TxPDO6*, Nd.25	494	0x1EE	TxPDO6*, Nd.46
453	0x1C5	TxPDO6*, Nd.5	474	0x1DA	TxPDO6*, Nd.26	495	0x1EF	TxPDO6*, Nd.47
454	0x1C6	TxPDO6*, Nd.6	475	0x1DB	TxPDO6*, Nd.27	496	0x1F0	TxPDO6*, Nd.48
455	0x1C7	TxPDO6*, Nd.7	476	0x1DC	TxPDO6*, Nd.28	497	0x1F1	TxPDO6*, Nd.49
456	0x1C8	TxPDO6*, Nd.8	477	0x1DD	TxPDO6*, Nd.29	498	0x1F2	TxPDO6*, Nd.50
457	0x1C9	TxPDO6*, Nd.9	478	0x1DE	TxPDO6*, Nd.30	499	0x1F3	TxPDO6*, Nd.51
458	0x1CA	TxPDO6*, Nd.10	479	0x1DF	TxPDO6*, Nd.31	500	0x1F4	TxPDO6*, Nd.52
459	0x1CB	TxPDO6*, Nd.11	480	0x1E0	TxPDO6*, Nd.32	501	0x1F5	TxPDO6*, Nd.53
460	0x1CC	TxPDO6*, Nd.12	481	0x1E1	TxPDO6*, Nd.33	502	0x1F6	TxPDO6*, Nd.54
461	0x1CD	TxPDO6*, Nd.13	482	0x1E2	TxPDO6*, Nd.34	503	0x1F7	TxPDO6*, Nd.55
462	0x1CE	TxPDO6*, Nd.14	483	0x1E3	TxPDO6*, Nd.35	504	0x1F8	TxPDO6*, Nd.56
463	0x1CF	TxPDO6*, Nd.15	484	0x1E4	TxPDO6*, Nd.36	505	0x1F9	TxPDO6*, Nd.57
464	0x1D0	TxPDO6*, Nd.16	485	0x1E5	TxPDO6*, Nd.37	506	0x1FA	TxPDO6*, Nd.58
465	0x1D1	TxPDO6*, Nd.17	486	0x1E6	TxPDO6*, Nd.38	507	0x1FB	TxPDO6*, Nd.59
466	0x1D2	TxPDO6*, Nd.18	487	0x1E7	TxPDO6*, Nd.39	508	0x1FC	TxPDO6*, Nd.60
467	0x1D3	TxPDO6*, Nd.19	488	0x1E8	TxPDO6*, Nd.40	509	0x1FD	TxPDO6*, Nd.61
468	0x1D4	TxPDO6*, Nd.20	489	0x1E9	TxPDO6*, Nd.41	510	0x1FE	TxPDO6*, Nd.62
469	0x1D5	TxPDO6*, Nd.21	490	0x1EA	TxPDO6*, Nd.42	511	0x1FF	TxPDO6*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
577	0x241	RxPDO6*, Nd.1	598	0x256	RxPDO6*, Nd.22	619	0x26B	RxPDO6* Nd.43
578	0x242	RxPDO6*, Nd.2	599	0x257	RxPDO6*, Nd.23	620	0x26C	RxPDO6, Nd.44
579	0x243	RxPDO6*, Nd.3	600	0x258	RxPDO6*, Nd.24	621	0x26D	RxPDO6*, Nd.45
580	0x244	RxPDO6*, Nd.4	601	0x259	RxPDO6*, Nd.25	622	0x26E	RxPDO6*, Nd.46
581	0x245	RxPDO6*, Nd.5	602	0x25A	RxPDO6*, Nd.26	623	0x26F	RxPDO6*, Nd.47
582	0x246	RxPDO6*, Nd.6	603	0x25B	RxPDO6*, Nd.27	624	0x270	RxPDO6*, Nd.48
583	0x247	RxPDO6*, Nd.7	604	0x25C	RxPDO6*, Nd.28	625	0x271	RxPDO6*, Nd.49
584	0x248	RxPDO6*, Nd.8	605	0x25D	RxPDO6*, Nd.29	626	0x272	RxPDO6*, Nd.50
585	0x249	RxPDO6*, Nd.9	606	0x25E	RxPDO6*, Nd.30	627	0x273	RxPDO6*, Nd.51
586	0x24A	RxPDO6*, Nd.10	607	0x25F	RxPDO6*, Nd.31	628	0x274	RxPDO6*, Nd.52
587	0x24B	RxPDO6*, Nd.11	608	0x260	RxPDO6*, Nd.32	629	0x275	RxPDO6*, Nd.53
588	0x24C	RxPDO6*, Nd.12	609	0x261	RxPDO6*, Nd.33	630	0x276	RxPDO6*, Nd.54
589	0x24D	RxPDO6*, Nd.13	610	0x262	RxPDO6*, Nd.34	631	0x277	RxPDO6*, Nd.55
590	0x24E	RxPDO6*, Nd.14	611	0x263	RxPDO6*, Nd.35	632	0x278	RxPDO6*, Nd.56
591	0x24F	RxPDO6*, Nd.15	612	0x264	RxPDO6*, Nd.36	633	0x279	RxPDO6*, Nd.57
592	0x250	RxPDO6*, Nd.16	613	0x265	RxPDO6*, Nd.3	634	0x27A	RxPDO6*, Nd.58
593	0x251	RxPDO6*, Nd.17	614	0x266	RxPDO6*, Nd.8	635	0x27B	RxPDO6*, Nd.59
594	0x252	RxPDO6*, Nd.18	615	0x267	RxPDO6*, Nd39	636	0x27C	RxPDO6*, Nd.60
595	0x253	RxPDO6*, Nd.19	616	0x268	RxPDO6*, N.40	637	0x27D	RxPDO6*, Nd.61
596	0x254	RxPDO6*, Nd.20	617	0x269	RxPDO6*, d.41	638	0x27E	RxPDO6*, Nd.62
597	0x255	RxPDO6*, Nd.21	618	0x26A	RxPDO6*,Nd.42	639	0x27F	RxPDO6*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
705	0x2C1	TxPDO7*, Nd.1	726	0x2D6	TxPDO7*, Nd.22	747	0x2EB	TxPDO7*, Nd.43
706	0x2C2	TxPDO7*, Nd.2	727	0x2D7	TxPDO7*, Nd.23	748	0x2EC	TxPDO7*, Nd.44
707	0x2C3	TxPDO7*, Nd.3	728	0x2D8	TxPDO7*, Nd.24	749	0x2ED	TxPDO7*, Nd.45
708	0x2C4	TxPDO7*, Nd.4	729	0x2D9	TxPDO7*, Nd.25	750	0x2EE	TxPDO7*, Nd.46
709	0x2C5	TxPDO7*, Nd.5	730	0x2DA	TxPDO7*, Nd.26	751	0x2EF	TxPDO7*, Nd.47
710	0x2C6	TxPDO7*, Nd.6	731	0x2DB	TxPDO7*, Nd.27	752	0x2F0	TxPDO7*, Nd.48
711	0x2C7	TxPDO7*, Nd.7	732	0x2DC	TxPDO7*, Nd.28	753	0x2F1	TxPDO7*, Nd.49
712	0x2C8	TxPDO7*, Nd.8	733	0x2DD	TxPDO7*, Nd.29	754	0x2F2	TxPDO7*, Nd.50
713	0x2C9	TxPDO7*, Nd.9	734	0x2DE	TxPDO7*, Nd.30	755	0x2F3	TxPDO7*, Nd.51
714	0x2CA	TxPDO7*, Nd.10	735	0x2DF	TxPDO7*, Nd.31	756	0x2F4	TxPDO7*, Nd.52
715	0x2CB	TxPDO7*, Nd.11	736	0x2E0	TxPDO7*, Nd.32	757	0x2F5	TxPDO7*, Nd.53
716	0x2CC	TxPDO7*, Nd.12	737	0x2E1	TxPDO7*, Nd.33	758	0x2F6	TxPDO7*, Nd.54
717	0x2CD	TxPDO7*, Nd.13	738	0x2E2	TxPDO7*, Nd.34	759	0x2F7	TxPDO7*, Nd.55
718	0x2CE	TxPDO7*, Nd.14	739	0x2E3	TxPDO7*, Nd.35	760	0x2F8	TxPDO7*, Nd.56
719	0x2CF	TxPDO7*, Nd.15	740	0x2E4	TxPDO7*, Nd.36	761	0x2F9	TxPDO7*, Nd.57
720	0x2D0	TxPDO7*, Nd.16	741	0x2E5	TxPDO7*, Nd.37	762	0x2FA	TxPDO7*, Nd.58
721	0x2D1	TxPDO7*, Nd.17	742	0x2E6	TxPDO7*, Nd.38	763	0x2FB	TxPDO7*, Nd.59
722	0x2D2	TxPDO7*, Nd.18	743	0x2E7	TxPDO7*, Nd.39	764	0x2FC	TxPDO7*, Nd.60
723	0x2D3	TxPDO7*, Nd.19	744	0x2E8	TxPDO7*, Nd.40	765	0x2FD	TxPDO7*, Nd.61
724	0x2D4	TxPDO7*, Nd.20	745	0x2E9	TxPDO7*, Nd.41	766	0x2FE	TxPDO7*, Nd.62
725	0x2D5	TxPDO7*, Nd.21	746	0x2EA	TxPDO7*, Nd.42	767	0x2FF	TxPDO7*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
833	0x341	RxPDO7*, Nd.1	854	0x356	RxPDO7*, Nd.22	875	0x36B	RxPDO7*, Nd.43
834	0x342	RxPDO7*, Nd.2	855	0x357	RxPDO7*, Nd.23	876	0x36C	RxPDO7*, Nd.44
835	0x343	RxPDO7*, Nd.3	856	0x358	RxPDO7*, Nd.24	877	0x36D	RxPDO7*, Nd.45
836	0x344	RxPDO7*, Nd.4	857	0x359	RxPDO7*, Nd.25	878	0x36E	RxPDO7*, Nd.46
837	0x345	RxPDO7*, Nd.5	858	0x35A	RxPDO7*, Nd.26	879	0x36F	RxPDO7*, Nd.47
838	0x346	RxPDO7*, Nd.6	859	0x35B	RxPDO7*, Nd.27	880	0x370	RxPDO7*, Nd.48
839	0x347	RxPDO7*, Nd.7	860	0x35C	RxPDO7*, Nd.28	881	0x371	RxPDO7*, Nd.49
840	0x348	RxPDO7*, Nd.8	861	0x35D	RxPDO7*, Nd.29	882	0x372	RxPDO7*, Nd.50
841	0x349	RxPDO7*, Nd.9	862	0x35E	RxPDO7*, Nd.30	883	0x373	RxPDO7*, Nd.51
842	0x34A	RxPDO7*, Nd.10	863	0x35F	RxPDO7*, Nd.31	884	0x374	RxPDO7*, Nd.52
843	0x34B	RxPDO7*, Nd.11	864	0x360	RxPDO7*, Nd.32	885	0x375	RxPDO7*, Nd.53
844	0x34C	RxPDO7*, Nd.12	865	0x361	RxPDO7*, Nd.33	886	0x376	RxPDO7*, Nd.54
845	0x34D	RxPDO7*, Nd.13	866	0x362	RxPDO7*, Nd.34	887	0x377	RxPDO7*, Nd.55
846	0x34E	RxPDO7*, Nd.14	867	0x363	RxPDO7*, Nd.35	888	0x378	RxPDO7*, Nd.56
847	0x34F	RxPDO7*, Nd.15	868	0x364	RxPDO7*, Nd.36	889	0x379	RxPDO7*, Nd.57
848	0x350	RxPDO7*, Nd.16	869	0x365	RxPDO7*, Nd.37	890	0x37A	RxPDO7*, Nd.58
849	0x351	RxPDO7*, Nd.17	870	0x366	RxPDO7*, Nd.38	891	0x37B	RxPDO7*, Nd.59
850	0x352	RxPDO7*, Nd.18	871	0x367	RxPDO7*, Nd.39	892	0x37C	RxPDO7*, Nd.60
851	0x353	RxPDO7*, Nd.19	872	0x368	RxPDO7*, Nd.40	893	0x37D	RxPDO7*, Nd.61
852	0x354	RxPDO7*, Nd.20	873	0x369	RxPDO7*, Nd.41	894	0x37E	RxPDO7*, Nd.62
853	0x355	RxPDO7*, Nd.21	874	0x36A	RxPDO7*, Nd.42	895	0x37F	RxPDO7*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
961	0x3C1	TxPDO8*, Nd.1	982	0x3D6	TxPDO8*, Nd.22	1003	0x3EB	TxPDO8*, Nd.43
962	0x3C2	TxPDO8*, Nd.2	983	0x3D7	TxPDO8*, Nd.23	1004	0x3EC	TxPDO8*, Nd.44
963	0x3C3	TxPDO8*, Nd.3	984	0x3D8	TxPDO8*, Nd.24	1005	0x3ED	TxPDO8*, Nd.45
964	0x3C4	TxPDO8*, Nd.4	985	0x3D9	TxPDO8*, Nd.25	1006	0x3EE	TxPDO8*, Nd.46
965	0x3C5	TxPDO8*, Nd.5	986	0x3DA	TxPDO8*, Nd.26	1007	0x3EF	TxPDO8*, Nd.47
966	0x3C6	TxPDO8*, Nd.6	987	0x3DB	TxPDO8*, Nd.27	1008	0x3F0	TxPDO8*, Nd.48
967	0x3C7	TxPDO8*, Nd.7	988	0x3DC	TxPDO8*, Nd.28	1009	0x3F1	TxPDO8*, Nd.49
968	0x3C8	TxPDO8*, Nd.8	989	0x3DD	TxPDO8*, Nd.29	1010	0x3F2	TxPDO8*, Nd.50
969	0x3C9	TxPDO8*, Nd.9	990	0x3DE	TxPDO8*, Nd.30	1011	0x3F3	TxPDO8*, Nd.51
970	0x3CA	TxPDO8*, Nd.10	991	0x3DF	TxPDO8*, Nd.31	1012	0x3F4	TxPDO8*, Nd.52
971	0x3CB	TxPDO8*, Nd.11	992	0x3E0	TxPDO8*, Nd.32	1013	0x3F5	TxPDO8*, Nd.53
972	0x3CC	TxPDO8*, Nd.12	993	0x3E1	TxPDO8*, Nd.33	1014	0x3F6	TxPDO8*, Nd.54
973	0x3CD	TxPDO8*, Nd.13	994	0x3E2	TxPDO8*, Nd.34	1015	0x3F7	TxPDO8*, Nd.55
974	0x3CE	TxPDO8*, Nd.14	995	0x3E3	TxPDO8*, Nd.35	1016	0x3F8	TxPDO8*, Nd.56
975	0x3CF	TxPDO8*, Nd.15	996	0x3E4	TxPDO8*, Nd.36	1017	0x3F9	TxPDO8*, Nd.57
976	0x3D0	TxPDO8*, Nd.16	997	0x3E5	TxPDO8*, Nd.37	1018	0x3FA	TxPDO8*, Nd.58
977	0x3D1	TxPDO8*, Nd.17	998	0x3E6	TxPDO8*, Nd.38	1019	0x3FB	TxPDO8*, Nd.59
978	0x3D2	TxPDO8*, Nd.18	999	0x3E7	TxPDO8*, Nd.39	1020	0x3FC	TxPDO8*, Nd.60
979	0x3D3	TxPDO8*, Nd.19	1000	0x3E8	TxPDO8*, Nd.40	1021	0x3FD	TxPDO8*, Nd.61
980	0x3D4	TxPDO8*, Nd.20	1001	0x3E9	TxPDO8*, Nd.41	1022	0x3FE	TxPDO8*, Nd.62
981	0x3D5	TxPDO8*, Nd.21	1002	0x3EA	TxPDO8*, Nd.42	1023	0x3FF	TxPDO8*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1089	0x441	RxPDO8*, Nd.1	1110	0x456	RxPDO8*, Nd.22	1131	0x46B	RxPDO8*, Nd.43
1090	0x442	RxPDO8*, Nd.2	1111	0x457	RxPDO8*, Nd.23	1132	0x46C	RxPDO8*, Nd.44
1091	0x443	RxPDO8*, Nd.3	1112	0x458	RxPDO8*, Nd.24	1133	0x46D	RxPDO8*, Nd.45
1092	0x444	RxPDO8*, Nd.4	1113	0x459	RxPDO8*, Nd.25	1134	0x46E	RxPDO8*, Nd.46
1093	0x445	RxPDO8*, Nd.5	1114	0x45A	RxPDO8*, Nd.26	1135	0x46F	RxPDO8*, Nd.47
1094	0x446	RxPDO8*, Nd.6	1115	0x45B	RxPDO8*, Nd.27	1136	0x470	RxPDO8*, Nd.48
1095	0x447	RxPDO8*, Nd.7	1116	0x45C	RxPDO8*, Nd.28	1137	0x471	RxPDO8*, Nd.49
1096	0x448	RxPDO8*, Nd.8	1117	0x45D	RxPDO8*, Nd.29	1138	0x472	RxPDO8*, Nd.50
1097	0x449	RxPDO8*, Nd.9	1118	0x45E	RxPDO8*, Nd.30	1139	0x473	RxPDO8*, Nd.51
1098	0x44A	RxPDO8*, Nd.10	1119	0x45F	RxPDO8*, Nd.31	1140	0x474	RxPDO8*, Nd.52
1099	0x44B	RxPDO8*, Nd.11	1120	0x460	RxPDO8*, Nd.32	1141	0x475	RxPDO8*, Nd.53
1100	0x44C	RxPDO8*, Nd.12	1121	0x461	RxPDO8*, Nd.33	1142	0x476	RxPDO8*, Nd.54
1101	0x44D	RxPDO8*, Nd.13	1122	0x462	RxPDO8*, Nd.34	1143	0x477	RxPDO8*, Nd.55
1102	0x44E	RxPDO8*, Nd.14	1123	0x463	RxPDO8*, Nd.35	1144	0x478	RxPDO8*, Nd.56
1103	0x44F	RxPDO8*, Nd.15	1124	0x464	RxPDO8*, Nd.36	1145	0x479	RxPDO8*, Nd.57
1104	0x450	RxPDO8*, Nd.16	1125	0x465	RxPDO8*, Nd.37	1146	0x47A	RxPDO8*, Nd.58
1105	0x451	RxPDO8*, Nd.17	1126	0x466	RxPDO8*, Nd.38	1147	0x47B	RxPDO8*, Nd.59
1106	0x452	RxPDO8*, Nd.18	1127	0x467	RxPDO8*, Nd.39	1148	0x47C	RxPDO8*, Nd.60
1107	0x453	RxPDO8*, Nd.19	1128	0x468	RxPDO8*, Nd.40	1149	0x47D	RxPDO8*, Nd.61
1108	0x454	RxPDO8*, Nd.20	1129	0x469	RxPDO8*, Nd.41	1150	0x47E	RxPDO8*, Nd.62
1109	0x455	RxPDO8*, Nd.21	1130	0x46A	RxPDO8*, Nd.42	1151	0x47F	RxPDO8*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1217	0x4C1	TxPDO9*, Nd.1	1238	0x4D6	TxPDO9*, Nd.22	1259	0x4EB	TxPDO9*, Nd.43
1218	0x4C2	TxPDO9*, Nd.2	1239	0x4D7	TxPDO9*, Nd.23	1260	0x4EC	TxPDO9*, Nd.44
1219	0x4C3	TxPDO9*, Nd.3	1240	0x4D8	TxPDO9*, Nd.24	1261	0x4ED	TxPDO9*, Nd.45
1220	0x4C4	TxPDO9*, Nd.4	1241	0x4D9	TxPDO9*, Nd.25	1262	0x4EE	TxPDO9*, Nd.46
1221	0x4C5	TxPDO9*, Nd.5	1242	0x4DA	TxPDO9*, Nd.26	1263	0x4EF	TxPDO9*, Nd.47
1222	0x4C6	TxPDO9*, Nd.6	1243	0x4DB	TxPDO9*, Nd.27	1264	0x4F0	TxPDO9*, Nd.48
1223	0x4C7	TxPDO9*, Nd.7	1244	0x4DC	TxPDO9*, Nd.28	1265	0x4F1	TxPDO9*, Nd.49
1224	0x4C8	TxPDO9*, Nd.8	1245	0x4DD	TxPDO9*, Nd.29	1266	0x4F2	TxPDO9*, Nd.50
1225	0x4C9	TxPDO9*, Nd.9	1246	0x4DE	TxPDO9*, Nd.30	1267	0x4F3	TxPDO9*, Nd.51
1226	0x4CA	TxPDO9*, Nd.10	1247	0x4DF	TxPDO9*, Nd.31	1268	0x4F4	TxPDO9*, Nd.52
1227	0x4CB	TxPDO9*, Nd.11	1248	0x4E0	TxPDO9*, Nd.32	1269	0x4F5	TxPDO9*, Nd.53
1228	0x4CC	TxPDO9*, Nd.12	1249	0x4E1	TxPDO9*, Nd.33	1270	0x4F6	TxPDO9*, Nd.54
1229	0x4CD	TxPDO9*, Nd.13	1250	0x4E2	TxPDO9*, Nd.34	1271	0x4F7	TxPDO9*, Nd.55
1230	0x4CE	TxPDO9*, Nd.14	1251	0x4E3	TxPDO9*, Nd.35	1272	0x4F8	TxPDO9*, Nd.56
1231	0x4CF	TxPDO9*, Nd.15	1252	0x4E4	TxPDO9*, Nd.36	1273	0x4F9	TxPDO9*, Nd.57
1232	0x4D0	TxPDO9*, Nd.16	1253	0x4E5	TxPDO9*, Nd.37	1274	0x4FA	TxPDO9*, Nd.58
1233	0x4D1	TxPDO9*, Nd.17	1254	0x4E6	TxPDO9*, Nd.38	1275	0x4FB	TxPDO9*, Nd.59
1234	0x4D2	TxPDO9*, Nd.18	1255	0x4E7	TxPDO9*, Nd.39	1276	0x4FC	TxPDO9*, Nd.60
1235	0x4D3	TxPDO9*, Nd.19	1256	0x4E8	TxPDO9*, Nd.40	1277	0x4FD	TxPDO9*, Nd.61
1236	0x4D4	TxPDO9*, Nd.20	1257	0x4E9	TxPDO9*, Nd.41	1278	0x4FE	TxPDO9*, Nd.62
1237	0x4D5	TxPDO9*, Nd.21	1258	0x4EA	TxPDO9*, Nd.42	1279	0x4FF	TxPDO9*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1345	0x541	RxPDO9*, Nd.1	1366	0x556	RxPDO9*, Nd.22	1387	0x56B	RxPDO9*, Nd.43
1346	0x542	RxPDO9*, Nd.2	1367	0x557	RxPDO9*, Nd.23	1388	0x56C	RxPDO9*, Nd.44
1347	0x543	RxPDO9*, Nd.3	1368	0x558	RxPDO9*, Nd.24	1389	0x56D	RxPDO9*, Nd.45
1348	0x544	RxPDO9*, Nd.4	1369	0x559	RxPDO9*, Nd.25	1390	0x56E	RxPDO9*, Nd.46
1349	0x545	RxPDO9*, Nd.5	1370	0x55A	RxPDO9*, Nd.26	1391	0x56F	RxPDO9*, Nd.47
1350	0x546	RxPDO9*, Nd.6	1371	0x55B	RxPDO9*, Nd.27	1392	0x570	RxPDO9*, Nd.48
1351	0x547	RxPDO9*, Nd.7	1372	0x55C	RxPDO9*, Nd.28	1393	0x571	RxPDO9*, Nd.49
1352	0x548	RxPDO9*, Nd.8	1373	0x55D	RxPDO9*, Nd.29	1394	0x572	RxPDO9*, Nd.50
1353	0x549	RxPDO9*, Nd.9	1374	0x55E	RxPDO9*, Nd.30	1395	0x573	RxPDO9*, Nd.51
1354	0x54A	RxPDO9*, Nd.10	1375	0x55F	RxPDO9*, Nd.31	1396	0x574	RxPDO9*, Nd.52
1355	0x54B	RxPDO9*, Nd.11	1376	0x560	RxPDO9*, Nd.32	1397	0x575	RxPDO9*, Nd.53
1356	0x54C	RxPDO9*, Nd.12	1377	0x561	RxPDO9*, Nd.33	1398	0x576	RxPDO9*, Nd.54
1357	0x54D	RxPDO9*, Nd.13	1378	0x562	RxPDO9*, Nd.34	1399	0x577	RxPDO9*, Nd.55
1358	0x54E	RxPDO9*, Nd.14	1379	0x563	RxPDO9*, Nd.35	1400	0x578	RxPDO9*, Nd.56
1359	0x54F	RxPDO9*, Nd.15	1380	0x564	RxPDO9*, Nd.36	1401	0x579	RxPDO9*, Nd.57
1360	0x550	RxPDO9*, Nd.16	1381	0x565	RxPDO9*, Nd.37	1402	0x57A	RxPDO9*, Nd.58
1361	0x551	RxPDO9*, Nd.17	1382	0x566	RxPDO9*, Nd.38	1403	0x57B	RxPDO9*, Nd.59
1362	0x552	RxPDO9*, Nd.18	1383	0x567	RxPDO9*, Nd.39	1404	0x57C	RxPDO9*, Nd.60
1363	0x553	RxPDO9*, Nd.19	1384	0x568	RxPDO9*, Nd.40	1405	0x57D	RxPDO9*, Nd.61
1364	0x554	RxPDO9*, Nd.20	1385	0x569	RxPDO9*, Nd.41	1406	0x57E	RxPDO9*, Nd.62
1365	0x555	RxPDO9*, Nd.21	1386	0x56A	RxPDO9*, Nd.42	1407	0x57F	RxPDO9*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1473	0x5C1	TxPDO10*, Nd.1	1494	0x5D6	TxPDO10*, Nd.22	1515	0x5EB	TxPDO10*, Nd.43
1474	0x5C2	TxPDO10*, Nd.2	1495	0x5D7	TxPDO10*, Nd.23	1516	0x5EC	TxPDO10*, Nd.44
1475	0x5C3	TxPDO10*, Nd.3	1496	0x5D8	TxPDO10*, Nd.24	1517	0x5ED	TxPDO10*, Nd.45
1476	0x5C4	TxPDO10*, Nd.4	1497	0x5D9	TxPDO10*, Nd.25	1518	0x5EE	TxPDO10*, Nd.46
1477	0x5C5	TxPDO10*, Nd.5	1498	0x5DA	TxPDO10*, Nd.26	1519	0x5EF	TxPDO10*, Nd.47
1478	0x5C6	TxPDO10*, Nd.6	1499	0x5DB	TxPDO10*, Nd.27	1520	0x5F0	TxPDO10*, Nd.48
1479	0x5C7	TxPDO10*, Nd.7	1500	0x5DC	TxPDO10*, Nd.28	1521	0x5F1	TxPDO10*, Nd.49
1480	0x5C8	TxPDO10*, Nd.8	1501	0xDE	TxPDO10*, Nd.29	1522	0x5F2	TxPDO10*, Nd.50
1481	0x5C9	TxPDO10*, Nd.9	1502	0x5DE	TxPDO10*, Nd.30	1523	0x5F3	TxPDO10*, Nd.51
1482	0x5CA	TxPDO10*, Nd.10	1503	0x5DF	TxPDO10*, Nd.31	1524	0x5F4	TxPDO10*, Nd.52
1483	0x5CB	TxPDO10*, Nd.11	1504	0x5E0	TxPDO10*, Nd.32	1525	0x5F5	TxPDO10*, Nd.53
1484	0x5CC	TxPDO10*, Nd.12	1505	0x5E1	TxPDO10*, Nd.33	1526	0x5F6	TxPDO10*, Nd.54
1485	0x5CD	TxPDO10*, Nd.13	1506	0x5E2	TxPDO10*, Nd.34	1527	0x5F7	TxPDO10*, Nd.55
1486	0x5CE	TxPDO10*, Nd.14	1507	0x5E3	TxPDO10*, Nd.35	1528	0x5F8	TxPDO10*, Nd.56
1487	0x5CF	TxPDO10*, Nd.15	1508	0x5E4	TxPDO10*, Nd.36	1529	0x5F9	TxPDO10*, Nd.57
1488	0x5D0	TxPDO10*, Nd.16	1509	0x5E5	TxPDO10*, Nd.37	1530	0x5FA	TxPDO10*, Nd.58
1489	0x5D1	TxPDO10*, Nd.17	1510	0x5E6	TxPDO10*, Nd.38	1531	0x5FB	TxPDO10*, Nd.59
1490	0x5D2	TxPDO10*, Nd.18	1511	0x5E7	TxPDO10*, Nd.39	1532	0x5FC	TxPDO10*, Nd.60
1491	0x5D3	TxPDO10*, Nd.19	1512	0x5E8	TxPDO10*, Nd.40	1533	0x5FD	TxPDO10*, Nd.61
1492	0x5D4	TxPDO10*, Nd.20	1513	0x5E9	TxPDO10*, Nd.41	1534	0x5FE	TxPDO10*, Nd.62
1493	0x5D5	TxPDO10*, Nd.21	1514	0x5EA	TxPDO10*, Nd.42	1535	0x5FF	TxPDO10*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1601	0x641	RxPDO10*, Nd.1	1622	0x656	RxPDO10*, Nd.22	1643	0x66B	RxPDO10*, Nd.43
1602	0x642	RxPDO10*, Nd.2	1623	0x657	RxPDO10*, Nd.23	1644	0x66C	RxPDO10*, Nd.44
1603	0x643	RxPDO10*, Nd.3	1624	0x658	RxPDO10*, Nd.24	1645	0x66D	RxPDO10*, Nd.45
1604	0x644	RxPDO10*, Nd.4	1625	0x659	RxPDO10*, Nd.25	1646	0x66E	RxPDO10*, Nd.46
1605	0x645	RxPDO10*, Nd.5	1626	0x65A	RxPDO10*, Nd.26	1647	0x66F	RxPDO10*, Nd.47
1606	0x646	RxPDO10*, Nd.6	1627	0x65B	RxPDO10*, Nd.27	1648	0x670	RxPDO10*, Nd.48
1607	0x647	RxPDO10*, Nd.7	1628	0x65C	RxPDO10*, Nd.28	1649	0x671	RxPDO10*, Nd.49
1608	0x648	RxPDO10*, Nd.8	1629	0x65D	RxPDO10*, Nd.29	1650	0x672	RxPDO10*, Nd.50
1609	0x649	RxPDO10*, Nd.9	1630	0x65E	RxPDO10*, Nd.30	1651	0x673	RxPDO10*, Nd.51
1610	0x64A	RxPDO10*, Nd.10	1631	0x65F	RxPDO10*, Nd.31	1652	0x674	RxPDO10*, Nd.52
1611	0x64B	RxPDO10*, Nd.11	1632	0x660	RxPDO10*, Nd.32	1653	0x675	RxPDO10*, Nd.53
1612	0x64C	RxPDO10*, Nd.12	1633	0x661	RxPDO10*, Nd.33	1654	0x676	RxPDO10*, Nd.54
1613	0x64D	RxPDO10*, Nd.13	1634	0x662	RxPDO10*, Nd.34	1655	0x677	RxPDO10*, Nd.55
1614	0x64E	RxPDO10*, Nd.14	1635	0x663	RxPDO10*, Nd.35	1656	0x678	RxPDO10*, Nd.56
1615	0x64F	RxPDO10*, Nd.15	1636	0x664	RxPDO10*, Nd.36	1657	0x679	RxPDO10*, Nd.57
1616	0x650	RxPDO10*, Nd.16	1637	0x665	RxPDO10*, Nd.37	1658	0x67A	RxPDO10*, Nd.58
1617	0x651	RxPDO10*, Nd.17	1638	0x666	RxPDO10*, Nd.38	1659	0x67B	RxPDO10*, Nd.59
1618	0x652	RxPDO10*, Nd.18	1639	0x667	RxPDO10*, Nd.39	1660	0x67C	RxPDO10*, Nd.60
1619	0x653	RxPDO10*, Nd.19	1640	0x668	RxPDO10*, Nd.40	1661	0x67D	RxPDO10*, Nd.61
1620	0x654	RxPDO10*, Nd.20	1641	0x669	RxPDO10*, Nd.41	1662	0x67E	RxPDO10*, Nd.62
1621	0x655	RxPDO10*, Nd.21	1642	0x66A	RxPDO10*, Nd.42	1663	0x67F	RxPDO10*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1729	0x6C1	TxPDO11*, Nd.1	1750	0x6D6	TxPDO11*, Nd.22	1771	0x6EB	TxPDO11*, Nd.43
1730	0x6C2	TxPDO11*, Nd.2	1751	0x6D7	TxPDO11*, Nd.23	1772	0x6EC	TxPDO11*, Nd.44
1731	0x6C3	TxPDO11*, Nd.3	1752	0x6D8	TxPDO11*, Nd.24	1773	0x6ED	TxPDO11*, Nd.45
1732	0x6C4	TxPDO11*, Nd.4	1753	0x6D9	TxPDO11*, Nd.25	1774	0x6EE	TxPDO11*, Nd.46
1733	0x6C5	TxPDO11*, Nd.5	1754	0x6DA	TxPDO11*, Nd.26	1775	0x6EF	TxPDO11*, Nd.47
1734	0x6C6	TxPDO11*, Nd.6	1755	0x6DB	TxPDO11*, Nd.27	1776	0x6F0	TxPDO11*, Nd.48
1735	0x6C7	TxPDO11*, Nd.7	1756	0x6DC	TxPDO11*, Nd.28	1777	0x6F1	TxPDO11*, Nd.49
1736	0x6C8	TxPDO11*, Nd.8	1757	0x6DD	TxPDO11*, Nd.29	1778	0x6F2	TxPDO11*, Nd.50
1737	0x6C9	TxPDO11*, Nd.9	1758	0x6DE	TxPDO11*, Nd.30	1779	0x6F3	TxPDO11*, Nd.51
1738	0x6CA	TxPDO11*, Nd.10	1759	0x6DF	TxPDO11*, Nd.31	1780	0x6F4	TxPDO11*, Nd.52
1739	0x6CB	TxPDO11*, Nd.11	1760	0x6E0	TxPDO11*, Nd.32	1781	0x6F5	TxPDO11*, Nd.53
1740	0x6CC	TxPDO11*, Nd.12	1761	0x6E1	TxPDO11*, Nd.33	1782	0x6F6	TxPDO11*, Nd.54
1741	0x6CD	TxPDO11*, Nd.13	1762	0x6E2	TxPDO11*, Nd.34	1783	0x6F7	TxPDO11*, Nd.55
1742	0x6CE	TxPDO11*, Nd.14	1763	0x6E3	TxPDO11*, Nd.35	1784	0x6F8	TxPDO11*, Nd.56
1743	0x6CF	TxPDO11*, Nd.15	1764	0x6E4	TxPDO11*, Nd.36	1785	0x6F9	TxPDO11*, Nd.57
1744	0x6D0	TxPDO11*, Nd.16	1765	0x6E5	TxPDO11*, Nd.37	1786	0x6FA	TxPDO11*, Nd.58
1745	0x6D1	TxPDO11*, Nd.17	1766	0x6E6	TxPDO11*, Nd.38	1787	0x6FB	TxPDO11*, Nd.59
1746	0x6D2	TxPDO11*, Nd.18	1767	0x6E7	TxPDO11*, Nd.39	1788	0x6FC	TxPDO11*, Nd.60
1747	0x6D3	TxPDO11*, Nd.19	1768	0x6E8	TxPDO11*, Nd.40	1789	0x6FD	TxPDO11*, Nd.61
1748	0x6D4	TxPDO11*, Nd.20	1769	0x6E9	TxPDO11*, Nd.41	1790	0x6FE	TxPDO11*, Nd.62
1749	0x6D5	TxPDO11*, Nd.21	1770	0x6EA	TxPDO11*, Nd.42	1791	0x6FF	TxPDO11*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1857	0x741	RxPDO11*, Nd.1	1878	0x756	RxPDO11*, Nd.22	1899	0x76B	RxPDO11*, Nd.43
1858	0x742	RxPDO11*, Nd.2	1879	0x757	RxPDO11*, Nd.23	1900	0x76C	RxPDO11*, Nd.44
1859	0x743	RxPDO11*, Nd.3	1880	0x758	RxPDO11*, Nd.24	1901	0x76D	RxPDO11*, Nd.45
1860	0x744	RxPDO11*, Nd.4	1881	0x759	RxPDO11*, Nd.25	1902	0x76E	RxPDO11*, Nd.46
1861	0x745	RxPDO11*, Nd.5	1882	0x75A	RxPDO11*, Nd.26	1903	0x76F	RxPDO11*, Nd.47
1862	0x746	RxPDO11*, Nd.6	1883	0x75B	RxPDO11*, Nd.27	1904	0x770	RxPDO11*, Nd.48
1863	0x747	RxPDO11*, Nd.7	1884	0x75C	RxPDO11*, Nd.28	1905	0x771	RxPDO11*, Nd.49
1864	0x748	RxPDO11*, Nd.8	1885	0x75D	RxPDO11*, Nd.29	1906	0x772	RxPDO11*, Nd.50
1865	0x749	RxPDO11*, Nd.9	1886	0x75E	RxPDO11*, Nd.30	1907	0x773	RxPDO11*, Nd.51
1866	0x74A	RxPDO11*, Nd.10	1887	0x75F	RxPDO11*, Nd.31	1908	0x774	RxPDO11*, Nd.52
1867	0x74B	RxPDO11*, Nd.11	1888	0x760	RxPDO11*, Nd.32	1909	0x775	RxPDO11*, Nd.53
1868	0x74C	RxPDO11*, Nd.12	1889	0x761	RxPDO11*, Nd.33	1910	0x776	RxPDO11*, Nd.54
1869	0x74D	RxPDO11*, Nd.13	1890	0x762	RxPDO11*, Nd.34	1911	0x777	RxPDO11*, Nd.55
1870	0x74E	RxPDO11*, Nd.14	1891	0x763	RxPDO11*, Nd.35	1912	0x778	RxPDO11*, Nd.56
1871	0x74F	RxPDO11*, Nd.15	1892	0x764	RxPDO11*, Nd.36	1913	0x779	RxPDO11*, Nd.57
1872	0x750	RxPDO11*, Nd.16	1893	0x765	RxPDO11*, Nd.37	1914	0x77A	RxPDO11*, Nd.58
1873	0x751	RxPDO11*, Nd.17	1894	0x766	RxPDO11*, Nd.38	1915	0x77B	RxPDO11*, Nd.59
1874	0x752	RxPDO11*, Nd.18	1895	0x767	RxPDO11*, Nd.39	1916	0x77C	RxPDO11*, Nd.60
1875	0x753	RxPDO11*, Nd.19	1896	0x768	RxPDO11*, Nd.40	1917	0x77D	RxPDO11*, Nd.61
1876	0x754	RxPDO11*, Nd.20	1897	0x769	RxPDO11*, Nd.41	1918	0x77E	RxPDO11*, Nd.62
1877	0x755	RxPDO11*, Nd.21	1898	0x76A	RxPDO11*, Nd.42	1919	0x77F	RxPDO11*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1409	0x581	SDO Tx Nd.1	1430	0x596	SDO Tx Nd.22	1451	0x5AB	SDO Tx Nd.43
1410	0x582	SDO Tx Nd.2	1431	0x597	SDO Tx Nd.23	1452	0x5AC	SDO Tx Nd.44
1411	0x583	SDO Tx Nd.3	1432	0x598	SDO Tx Nd.24	1453	0x5AD	SDO Tx Nd.45
1412	0x584	SDO Tx Nd.4	1433	0x599	SDO Tx Nd.25	1454	0x5AE	SDO Tx Nd.46
1413	0x585	SDO Tx Nd.5	1434	0x59A	SDO Tx Nd.26	1455	0x5AF	SDO Tx Nd.47
1414	0x586	SDO Tx Nd.6	1435	0x59B	SDO Tx Nd.27	1456	0x5B0	SDO Tx Nd.48
1415	0x587	SDO Tx Nd.7	1436	0x59C	SDO Tx Nd.28	1457	0x5B1	SDO Tx Nd.49
1416	0x588	SDO Tx Nd.8	1437	0x59D	SDO Tx Nd.29	1458	0x5B2	SDO Tx Nd.50
1417	0x589	SDO Tx Nd.9	1438	0x59E	SDO Tx Nd.30	1459	0x5B3	SDO Tx Nd.51
1418	0x58A	SDO Tx Nd.10	1439	0x59F	SDO Tx Nd.31	1460	0x5B4	SDO Tx Nd.52
1419	0x58B	SDO Tx Nd.11	1440	0x5A0	SDO Tx Nd.32	1461	0x5B5	SDO Tx Nd.53
1420	0x58C	SDO Tx Nd.12	1441	0x5A1	SDO Tx Nd.33	1462	0x5B6	SDO Tx Nd.54
1421	0x58D	SDO Tx Nd.13	1442	0x5A2	SDO Tx Nd.34	1463	0x5B7	SDO Tx Nd.55
1422	0x58E	SDO Tx Nd.14	1443	0x5A3	SDO Tx Nd.35	1464	0x5B8	SDO Tx Nd.56
1423	0x58F	SDO Tx Nd.15	1444	0x5A4	SDO Tx Nd.36	1465	0x5B9	SDO Tx Nd.57
1424	0x590	SDO Tx Nd.16	1445	0x5A5	SDO Tx Nd.37	1466	0x5BA	SDO Tx Nd.58
1425	0x591	SDO Tx Nd.17	1446	0x5A6	SDO Tx Nd.38	1467	0x5BB	SDO Tx Nd.59
1426	0x592	SDO Tx Nd.18	1447	0x5A7	SDO Tx Nd.39	1468	0x5BC	SDO Tx Nd.60
1427	0x593	SDO Tx Nd.19	1448	0x5A8	SDO Tx Nd.40	1469	0x5BD	SDO Tx Nd.61
1428	0x594	SDO Tx Nd.20	1449	0x5A9	SDO Tx Nd.41	1470	0x5BE	SDO Tx Nd.62
1429	0x595	SDO Tx Nd.21	1450	0x5AA	SDO Tx Nd.42	1471	0x5BF	SDO Tx Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1537	0x601	SDO Rx Nd.1	1558	0x616	SDO Rx Nd.22	1579	0x62B	SDO Rx Nd.43
1538	0x602	SDO Rx Nd.2	1559	0x617	SDO Rx Nd.23	1580	0x62C	SDO Rx Nd.44
1539	0x603	SDO Rx Nd.3	1560	0x618	SDO Rx Nd.24	1581	0x62D	SDO Rx Nd.45
1540	0x604	SDO Rx Nd.4	1561	0x619	SDO Rx Nd.25	1582	0x62E	SDO Rx Nd.46
1541	0x605	SDO Rx Nd.5	1562	0x61A	SDO Rx Nd.26	1583	0x62F	SDO Rx Nd.47
1542	0x606	SDO Rx Nd.6	1563	0x61B	SDO Rx Nd.27	1584	0x630	SDO Rx Nd.48
1543	0x607	SDO Rx Nd.7	1564	0x61C	SDO Rx Nd.28	1585	0x631	SDO Rx Nd.49
1544	0x608	SDO Rx Nd.8	1565	0x61D	SDO Rx Nd.29	1586	0x632	SDO Rx Nd.50
1545	0x609	SDO Rx Nd.9	1566	0x61E	SDO Rx Nd.30	1587	0x633	SDO Rx Nd.51
1546	0x60A	SDO Rx Nd.10	1567	0x61F	SDO Rx Nd.31	1588	0x634	SDO Rx Nd.52
1547	0x60B	SDO Rx Nd.11	1568	0x620	SDO Rx Nd.32	1589	0x635	SDO Rx Nd.53
1548	0x60C	SDO Rx Nd.12	1569	0x621	SDO Rx Nd.33	1590	0x636	SDO Rx Nd.54
1549	0x60D	SDO Rx Nd.13	1570	0x622	SDO Rx Nd.34	1591	0x637	SDO Rx Nd.55
1550	0x60E	SDO Rx Nd.14	1571	0x623	SDO Rx Nd.35	1592	0x638	SDO Rx Nd.56
1551	0x60F	SDO Rx Nd.15	1572	0x624	SDO Rx Nd.36	1593	0x639	SDO Rx Nd.57
1552	0x610	SDO Rx Nd.16	1573	0x625	SDO Rx Nd.37	1594	0x63A	SDO Rx Nd.58
1553	0x611	SDO Rx Nd.17	1574	0x626	SDO Rx Nd.38	1595	0x63B	SDO Rx Nd.59
1554	0x612	SDO Rx Nd.18	1575	0x627	SDO Rx Nd.39	1596	0x63C	SDO Rx Nd.60
1555	0x613	SDO Rx Nd.19	1576	0x628	SDO Rx Nd.40	1597	0x63D	SDO Rx Nd.61
1556	0x614	SDO Rx Nd.20	1577	0x629	SDO Rx Nd.41	1598	0x63E	SDO Rx Nd.62
1557	0x615	SDO Rx Nd.21	1578	0x62A	SDO Rx Nd.42	1599	0x63F	SDO Rx Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1793	0x701	Guarding Nd.1	1814	0x716	Guarding Nd.22	1835	0x72B	Guarding Nd.43
1794	0x702	Guarding Nd.2	1815	0x717	Guarding Nd.23	1836	0x72C	Guarding Nd.44
1795	0x703	Guarding Nd.3	1816	0x718	Guarding Nd.24	1837	0x72D	Guarding Nd.45
1796	0x704	Guarding Nd.4	1817	0x719	Guarding Nd.25	1838	0x72E	Guarding Nd.46
1797	0x705	Guarding Nd.5	1818	0x71A	Guarding Nd.26	1839	0x72F	Guarding Nd.47
1798	0x706	Guarding Nd.6	1819	0x71B	Guarding Nd.27	1840	0x730	Guarding Nd.48
1799	0x707	Guarding Nd.7	1820	0x71C	Guarding Nd.28	1841	0x731	Guarding Nd.49
1800	0x708	Guarding Nd.8	1821	0x71D	Guarding Nd.29	1842	0x732	Guarding Nd.50
1801	0x709	Guarding Nd.9	1822	0x71E	Guarding Nd.30	1843	0x733	Guarding Nd.51
1802	0x70A	Guarding Nd.10	1823	0x71F	Guarding Nd.31	1844	0x734	Guarding Nd.52
1803	0x70B	Guarding Nd.11	1824	0x720	Guarding Nd.32	1845	0x735	Guarding Nd.53
1804	0x70C	Guarding Nd.12	1825	0x721	Guarding Nd.33	1846	0x736	Guarding Nd.54
1805	0x70D	Guarding Nd.13	1826	0x722	Guarding Nd.34	1847	0x737	Guarding Nd.55
1806	0x70E	Guarding Nd.14	1827	0x723	Guarding Nd.35	1848	0x738	Guarding Nd.56
1807	0x70F	Guarding Nd.15	1828	0x724	Guarding Nd.36	1849	0x739	Guarding Nd.57
1808	0x710	Guarding Nd.16	1829	0x725	Guarding Nd.37	1850	0x73A	Guarding Nd.58
1809	0x711	Guarding Nd.17	1830	0x726	Guarding Nd.38	1851	0x73B	Guarding Nd.59
1810	0x712	Guarding Nd.18	1831	0x727	Guarding Nd.39	1852	0x73C	Guarding Nd.60
1811	0x713	Guarding Nd.19	1832	0x728	Guarding Nd.40	1853	0x73D	Guarding Nd.61
1812	0x714	Guarding Nd.20	1833	0x729	Guarding Nd.41	1854	0x73E	Guarding Nd.62
1813	0x715	Guarding Nd.21	1834	0x72A	Guarding Nd.42	1855	0x73F	Guarding Nd.63

8.2 Zulassungen

(in Vorbereitung)

8.3 Literaturverzeichnis

Deutsche Bücher

- Holger Zeltwander (Hrsg.):
  CANopen,
  VDE Verlag, 2001.197 Seiten,
  ISBN 3-800-72448-0

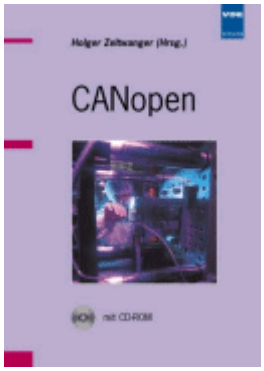


Abb. 62: CANopen

- Konrad Etschberger:
  Controller Area Network, Grundlagen, Protokolle, Bausteine, Anwendungen.
  Hanser Verlag, 2000. 431 Seiten.
  ISBN 3-446-19431-2

Feldbustechnik allgemein

- Gerhard Gruhler (Hrsg.):
  Feldbusse und Geräte-Kommunikationssysteme, Praktisches Know-How mit Vergleichsmöglichkeiten.
  Franzis Verlag 2001. 244 Seiten.
  ISBN 3-7723-5745-8

Englische Bücher

- Konrad Etschberger:
  Controller Area Network,
  Ixxat Press, 2001. 440 Seiten.
  ISBN 3-00-007376-0
- M. Farsi, M. Barbosa:
  CANopen Implementation,
  RSP 2000. 210 Seiten.
  ISBN 0-86380-247-8

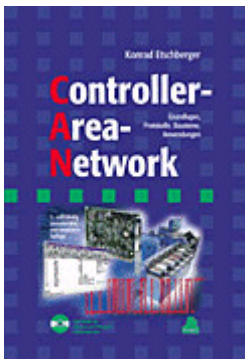


Abb. 63: Controller Area Network

Standards

- ISO 11898:
  Road Vehicles - Interchange of digital information - Controller Area Network (CAN) for high speed communication.
- CiA DS 301:
  CANopen Application Layer and Communication Profile.
  Erhältlich beim Verband CAN in Automation.
- CiA DS 401:
  CANopen Device Profile for Generic E/A Modules.
  Erhältlich beim Verband CAN in Automation.

8.4 Abkürzungsverzeichnis

CAN

Controller Area Network. In ISO 11898 standardisiertes serielles Bussystem, das als Basistechnologie für CANopen dient.

CiA

CAN in Automation e.V.. Internationaler Hersteller- und Nutzerverband mit Sitz in Erlangen/Deutschland.

COB

Communication Object. CAN-Telegramm mit bis zu 8 Datenbytes.

COB-ID

Communication Object Identifier. Telegrammadresse (nicht zu verwechseln mit Knotenadresse). CANopen verwendet die 11-Bit Identifier nach CAN 2.0A.

NMT

Network Management. Eines der Dienstelemente der CANopen-Spezifikation. Das Netzwerkmanagement dient zur Netzwerkinitialisierung und zur Knotenüberwachung.

PDO

Process Data Object oder Prozessdatenobjekt. CAN-Telegramm zur Übertragung von Prozessdaten (z. B. E/A-Daten).

RxPDO

Empfangs-PDO. PDOs werden immer aus Sicht des jeweiligen Gerätes bezeichnet. So wird ein TxPDO mit Eingangsdaten einer E/A Baugruppe zum RxPDO aus Sicht der Steuerung.

SDO

Service Data Object oder Servicedatenobjekt. CAN-Telegramm mit Protokoll zur Kommunikation mit Daten des Objektverzeichnisses (typisch Parameterdaten).

TxPDO

Sende-PDO (aus Sicht des CAN-Knotens bezeichnet).

8.5 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline:	+49(0)5246/963-157
Fax:	+49(0)5246/963-9157
E-Mail:	support@beckhoff.com

Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline:	+49(0)5246/963-460
Fax:	+49(0)5246/963-479
E-Mail:	service@beckhoff.com

Weitere Support- und Serviceadressen finden Sie auf unseren Internetseiten unter http://www.beckhoff.de.

Beckhoff Firmenzentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon:	+49(0)5246/963-0
Fax:	+49(0)5246/963-198
E-Mail:	info@beckhoff.com

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unseren Internetseiten:
http://www.beckhoff.de

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Abbildungsverzeichnis